



TUGAS AKHIR - TE 145561

**MONITORING SENSOR MESIN INDUSTRI DI AREA DIVISI
KAPAL NIAGA PT PAL SURABAYA MENGGUNAKAN APLIKASI
IOT BERBASIS JARINGAN INTERNET *WIFI***

Audy Zahra Firdausy
NRP 2214039014

Dosen Pembimbing
Eko Setijadi, ST., MT., Ph.D
Onie Meiyanto, S.Pd.

PROGRAM STUDI ELEKTRONIKA INDUSTRI
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***THE MONITORING OF INDUSTRIAL ENGINE SENSOR AT
THE KAPAL NIAGA DIVISION PT PAL SURABAYA USING IOT
APPLICATION BASED ON INTERNET WIFI NETWORK***

Audy Zahra Firdausy
NRP 2214039014

Supervisor
Eko Setijadi, ST., MT., Ph.D
Onie Meiyanto, S.Pd.

***INDUSTRIAL ELECTRONICS STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Monitoring Sensor Mesin Industri di Area Divisi Kapal Niaga PT PAL Surabaya Menggunakan Aplikasi IoT Berbasis Jaringan Internet *Wifi*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 20 Juli 2017



Audy Zahra Firdausy
NRP 2214039014

-----Halaman ini sengaja dikosongkan-----

**MONITORING SENSOR MESIN INDUSTRI DI AREA DIVISI
KAPAL NIAGA PT PAL SURABAYA MENGGUNAKAN
APLIKASI IOT BERBASIS JARINGAN INTERNET WIFI**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Elektronika Industri
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Eko Setijadi, ST., MT., Ph.D
NIP. 197210012003121002

Onie Meivanto, S.Pd
NIP. 198505012011011008

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

MONITORING SENSOR MESIN INDUSTRI DI AREA DIVISI KAPAL NIAGA PT PAL SURABAYA MENGGUNAKAN APLIKASI IOT BERBASIS JARINGAN INTERNET WIFI

Nama : Audy Zahra Firdausy
NRP : 2214039014
Dosen Pembimbing I : Eko Setijadi, ST., MT., Ph. D
NIP : 197210012003121002
Dosen Pembimbing II : Onnie Meiyanto, S.Pd.
NIP : 198505012011011008

ABSTRAK

Di dunia industri, mesin yang digunakan menghasilkan produksi dengan skala besar. Dari hasil produksi dengan skala besar ini, memungkinkan terjadi kurang terkontrolnya mesin dari proses produksi tersebut. Seperti contohnya, proses produksi pada mesin bending yang ada di PT PAL Surabaya. Oleh karena itu, pada Tugas Akhir ini penulis mengambil judul monitoring sensor mesin industri di area divisi kapal niaga PT PAL Surabaya menggunakan aplikasi IoT berbasis jaringan internet *wifi* dengan menggunakan sensor *proximity* tipe induktif yang berfungsi untuk mengetahui keadaan *switch* suatu mesin bending jika mengenai besi. Adapun sensor lain yang digunakan, sensor *accelerometer* yang berfungsi untuk mengetahui kemiringan sudut pipa besi yang dibengkokkan pada mesin bending. Pengendalian Sistem monitoring ini menggunakan komunikasi *wifi* agar dapat dimonitoring dari jarak jauh menggunakan *android*.

Dari hasil pengujian sensor memiliki rata-rata *error* yakni sebesar 4,5% pada sumbu X dan pada sumbu Y terdapat rata-rata *error* sebesar 6,6%. Sistem monitoring ini menggunakan komunikasi internet *wifi* yang dapat dimonitoring dari jarak jauh dengan adanya koneksi internet. pengujian lamanya kecepatan pengiriman data juga memiliki nilai *error* yakni *delay* 15 detik sebesar 78,6%, *delay* 20 detik sebesar 82,5%, dan *delay* 25 detik sebesar 86%. *Error* dari kecepatan pengiriman data tersebut disebabkan oleh jarak/ jangkauan *wifi*, koneksi *wifi*, serta ada dan tidak adanya suatu penghalang.

Kata Kunci : *proximity, accelerometer, wifi, android, switch*

-----Halaman ini sengaja dikosongkan-----

***The Monitoring of Industrial Engine Sensor At the Kapal Niaga
Division PT PAL Surabaya Using IoT Application Based On Internet
Wifi Network***

<i>Name</i>	: Audy Zahra Firdausy
<i>Registrasion Number</i>	: 2214039014
<i>Supervisor I</i>	: Eko Setijadi, ST., MT., Ph. D
<i>ID Number</i>	: 197210012003121002
<i>Supervisor II</i>	: Onnie Meiyanto, S.Pd.
<i>ID Number</i>	: 198505012011011008

ABSTRACT

In the industrial world, machines used to produce large-scale production. From the results of this large-scale production, allowing less controlled machines from the production process. For example, the production process on the bending machine in PT PAL Surabaya. Therefore, in this Final Project the author took the title monitoring industrial engine sensors diaArea division of commercial ships PT PAL Surabaya using IoT application based internet wifi network using inductive proximity sensor type that serves to determine the state of a bending machine switch when it comes to iron. As for other sensors used, accelerometer sensor that serves to know the angle of iron pipe bent at bending machine. Control This monitoring system uses wifi communications to be monitored remotely using android.

From the test of the sensor has an average error of 4.5% on the X axis and on the Y axis there is an average error of 6.6%. This monitoring system uses internet wifi communications that can be monitored remotely with the internet connection. Testing the duration of data transmission speed also has error value that is 15 seconds delay of 78,6%, delay 20 second equal to 82,5%, and delay 25 second equal to 86%. Error of data transmission speed is caused by the distance/ range of wifi, wifi connection, and the presence and absence of a barrier.

Keywords : proximity, accelerometer, wifi, android, switch

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Elektronika Industri, Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

MONITORING SENSOR MESIN INDUSTRI DI AREA DIVISI KAPAL NIAGA PT PAL SURABAYA MENGGUNAKAN APLIKASI IOT BERBASIS JARINGAN INTERNET WIFI

Dalam Tugas Akhir ini menggunakan sensor *proximity* untuk mengetahui keadaan *switch* dari suatu mesin bending diidentifikasi dengan ON/OFF dan sensor *accelerometer* untuk mengetahui kemiringan sudut pipa besi yang dibengkokkan pada mesin bending, diidentifikasi dengan koordinat X,Y, Z. Unit pengolah menggunakan mikrokontroler yang dihubungkan dengan *Wifi* agar dapat dimonitoring dan dikontrol dari jarak jauh menggunakan *android*.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Eko Setijadi, ST., MT., Ph.D dan Bapak Onnie Meiyanto, S.Pd. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, Elektro Industri angkatan 18 yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, **20 Juli 2017**

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xxi
 BAB I.....	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	4
1.7 Relevansi.....	5
 BAB II.....	 7
2.1 Tinjauan Pustaka.....	7
2.2 <i>Internet of Things</i>	7
2.2.1 Aplikasi <i>Server Internet of Things</i>	8
2.3 <i>Web</i>	9
2.4 Aplikasi <i>Android</i>	9
2.5 MIT APP <i>Invertor</i>	10
2.6 Arduino	10
2.6.1 Spesifikasi Arduino UNO R3	10
2.7 ESP8266 <i>Wifi Shield Tipe 12E</i>	12
2.8 Sensor <i>Proximity</i> Induktif.....	14
2.9 Perhitungan Perancangan Pembagi Tegangan	14
2.10 Sensor <i>Accelerometer</i>	15
2.11 Lampu Indikator.....	15
2.12 <i>Relay Module</i>	16
2.13 LCD (<i>Liquid Crystal Display</i>)	17

BAB III PERANCANGAN DAN PEMBUATAN ALAT.....	19
3.1 Diagram Blok Skenario Kerja Alat.....	19
3.2 Implementasi dan Konfigurasi <i>Wifi Shield</i>	22
3.2.1 Proses Konfigurasi <i>Server IoT</i>	22
3.3 Implementasi dan Konfigurasi Aplikasi <i>Android</i>	25
3.3.1 Desain pada APP <i>Invertor</i>	25
3.3.2 Membuat Tampilan <i>Login Blok</i> pada APP <i>Invertor</i>	28
3.4 Lokasi Penerapan.....	30
3.5 Perancangan <i>Hardware</i>	31
3.6 Perancangan <i>Software</i>	34
BAB IV PENGUJIAN DAN ANALISA.....	39
4.1 Pengujian Koneksi <i>Wifi Shield</i>	39
4.2 Pengujian Kecepatan Pengiriman Data Sensor di Jaringan Internet <i>Wifi</i>	41
4.3 Pengujian Sensor	50
4.4 Pengujian <i>Software ThingSpeak</i>	51
4.5 Pengujian Keseluruhan	57
BAB V PENUTUP.....	65
5.1 Kesimpulan.....	65
5.2 Saran	65
DAFTAR PUSTAKA.....	67
LAMPIRAN A.....	69
A.1 <i>Listing Program</i> pada Arduino	69
A.2 <i>Listing Diagram Blok MIT APP Invertor</i>	75
LAMPIRAN B	77
B.1 <i>Datasheet</i> Arduino Uno.....	77
B.2 <i>Datasheet</i> ESP8266 <i>wifi shield</i>	79
B.3 <i>Datasheet</i> LCD 16x2.....	83
LAMPIRAN C	85
C.1 Pengujian Kecepatan Pengiriman Data	85
DAFTAR RIWAYAT HIDUP.....	87

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Cakupan <i>Internet of Things</i>	8
Gambar 2.2 Arduino UNO R3	11
Gambar 2.3 ESP8266 <i>Wifi Shield</i> Tipe 12E	12
Gambar 2.4 Sensor <i>Proximity</i> Induktif.....	14
Gambar 2.5 Rangkaian Pembagi Tegangan untuk Sensor <i>Proximity</i> Induktif	14
Gambar 2.6 Sensor <i>Accelerometer</i>	15
Gambar 2.7 Lampu Indikator	15
Gambar 2.8 <i>Relay Module</i>	16
Gambar 2.9 LCD	17
Gambar 3.1 Diagram Blok Sistem.....	20
Gambar 3.2 <i>Sign Up ThingSpeak</i>	22
Gambar 3.3 <i>Login Thingspeak</i>	23
Gambar 3.4 Membuat <i>Channel</i> di <i>ThingSpeak</i>	23
Gambar 3.5 Konfigurasi <i>Channel</i> di <i>Thingspeak</i>	24
Gambar 3.6 <i>API Key</i>	24
Gambar 3.7 Membuat Desain Tampilan <i>Screen 1</i> pada MIT App <i>Invertor</i>	25
Gambar 3.8 Membuat Desain Tampilan <i>Screen 2</i> pada MIT App <i>Invertor</i>	26
Gambar 3.9 Membuat Desain Tampilan <i>Screen 3,4 dan 5</i> pada MIT App <i>Invertor</i>	27
Gambar 3.10 Membuat Blok Tampilan <i>Screen 1</i> pada MIT APP <i>Invertor</i>	28
Gambar 3.11 Membuat Blok Tampilan <i>Screen 2</i> pada MIT APP <i>Invertor</i>	28
Gambar 3.12 Membuat Blok Tampilan <i>Screen 3,4 dan 5</i> pada MIT APP <i>Invertor</i>	29
Gambar 3.13 Lokasi Penerapan Jaringan <i>Wifi</i> dan <i>Ethernet</i> di PT PAL Surabaya	30
Gambar 3.14 Denah Lokasi Jaringan <i>Wifi</i> dan <i>Ethernet</i> di PT PAL Surabaya.....	30
Gambar 3.15 Rancangan Bagian Pintu Panel	31
Gambar 3.16 Rancangan Posisi Sensor <i>Accelerometer</i> Pada Mekanik Bagian Dalam Kotak Panel.....	32

Gambar 3.17	Rancangan Posisi Sensor <i>Proximity</i> pada Mekanik Bagian Dalam Kotak Panel	33
Gambar 3.18	<i>Flowchart</i> Status.....	34
Gambar 3.19	<i>Flowchart</i> Pembacaan Data Sensor	36
Gambar 4.1	Tampilan Pengujian IP Address <i>Personal Computer</i>	40
Gambar 4.2	Tampilan Pengujian Koneksi antara <i>Personal Computer</i> dan ESP8266 <i>Wifi Shield</i>	40
Gambar 4.3	Tampilan Pengujian Kedua Koneksi antara <i>Personal Computer</i> dan <i>Router Wifi</i>	41
Gambar 4.4	Hasil Pengiriman Sensor pada <i>ThingSpeak</i> dengan Delay 15 detik.....	44
Gambar 4.5	Hasil Pengiriman Sensor pada <i>ThingSpeak</i> dengan Delay 20 detik.....	46
Gambar 4.6	Hasil Pengiriman Sensor pada <i>ThingSpeak</i> dengan Delay 25 detik.....	48
Gambar 4.7	Rangkaian Pengujian Sensor MPU6050	50
Gambar 4.8	Hasil Pengiriman Data Sensor di Sumbu X pada <i>ThingSpeak</i>	52
Gambar 4.9	Hasil Pengiriman Data Sensor di Sumbu Y pada <i>ThingSpeak</i>	53
Gambar 4.10	Hasil Pengiriman Data Sensor di Sumbu X pada <i>ThingSpeak</i>	54
Gambar 4.11	Hasil Pengiriman Data Sensor di Sumbu Y pada <i>ThingSpeak</i>	55
Gambar 4.12	Hasil Pengiriman Data Sensor di Sumbu X pada <i>ThingSpeak</i>	56
Gambar 4.13	Hasil Pengiriman Data Sensor di Sumbu X pada <i>ThingSpeak</i>	57
Gambar 4.14	Lampu Indikator Hijau Menyala saat Sistem Bekerja	58
Gambar 4.15	Lampu Indikator Merah Menyala saat Pengiriman Data Sensor.....	59
Gambar 4.16	Lampu Indikator Kuning Menyala saat Pengiriman Data Sensor.....	60
Gambar 4.17	Kedua Lampu Indikator Menyala saat Pengiriman Data Sensor.....	61
Gambar 4.18	Pembacaan Data pada <i>ThingSpeak</i> saat Data Sensor Terkirim.....	61
Gambar 4.19	Pembacaan Data pada <i>ThingSpeak</i> saat Data Sensor Terkirim.....	62

Gambar 4.20	Pembacaan Data pada <i>ThingSpeak</i> saat Data Sensor Terkirim.....	62
Gambar 4.21	Pembacaan Data pada <i>ThingSpeak</i> saat Data Sensor Terkirim.....	63
Gambar 4.22	Tampilan Monitoring Pembacaan Data Sensor di ThingSpeak	63
Gambar C.1	Pembacaan Sensor di ThingSpeak	85
Gambar C.2	Penghitungan Lamanya Waktu Pembacaan Data antara <i>Thingspeak</i> dan <i>Serial Monitor</i>	85

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

	HALAMAN
Table 2.1 Spesifikasi Arduino UNO R3	11
Table 2.2 Spesifikasi ESP8266 <i>Wifi Shield</i> Tipe 12E.....	13
Table 2.3 Keterangan Port pada ESP8266 <i>Wifi Shield</i>	13
Table 2.4 Spesifikasi Lampu Indikator	16
Table 4.1 Pengujian Sensor di Jaringan Internet <i>Wifi</i>	41
Table 4.2 Pewaktu Pengiriman Data pada <i>Serial Monitor</i>	42
Table 4.3 Pewaktu Pengiriman Data pada <i>Serial Monitor</i>	45
Table 4.4 Pewaktu Pengiriman Data pada <i>Serial Monitor</i>	47
Table 4.5 Data <i>Error</i> Pewaktu Pembacaan Sensor pada <i>Serial Monitor</i>	49
Table 4.6 Hasil <i>Prosentase</i> dari <i>Error</i> Pengiriman Data	49
Table 4.7 Hasil Pengujian Data Sensor Sumbu X	50
Table 4.8 Hasil Pengujian Data Sensor Sumbu Y	51

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era modern sekarang khususnya kebutuhan di bidang industri dalam pembuatan produk dapat terintegrasi dengan cepat dan terkontrol. Boleh dikatakan, tanpa adanya kemajuan teknologi tidak bisa membuat segala sesuatunya menjadi efisien. Seperti yang diketahui, sebuah perusahaan di bidang industri masih terdapat faktor-faktor yang kurang mendukung dalam melakukan proses produksi. Hal ini dapat mempengaruhi kualitas dari suatu produk tersebut. Oleh sebab itu, suatu perusahaan yang bergerak di bidang industri memanfaatkan adanya konsep IoT. *Internet of Things* (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT sendiri telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical system* (MEMS), dan Internet. Dalam mengkomunikasikan pembacaan sensor dibutuhkan *Internet of Things* agar dapat dikoneksikan melalui tampilan menu monitoring pada aplikasi *android* dan *Personal Computer Home*.

Oleh karena itu, diharapkan dengan adanya monitoring mesin yang ada di industri salah satu contohnya di PT PAL Surabaya dapat memperlancar dan mengurangi kecurangan dalam melakukan proses produksi. Dalam hal ini juga dapat membantu pengefisienan dalam pengontrolan mesin-mesin yang ada di industri.

Dalam Tugas Akhir ini monitoring sensor mesin industri yang akan dirancang menggunakan modul arduino ESP8266 sehingga operator mesin dapat memantau mesin meskipun tidak sedang berada di dalam kawasan PT PAL Surabaya. Pengontrol arduino dikoneksikan melalui *wifi*, dengan demikian monitoring sensor berbasis IoT dapat terhubung dari jarak jauh dengan menggunakan web secara *online* sebagai akses *login* dan menu *display* yang dibuat dari APP *Invertor* untuk melihat tampilan monitoring dari mesin.

Dengan melihat kondisi tersebut dapat diambil judul “Monitoring Sensor Mesin Industri Di Area Divisi Kapal Niaga PT PAL Surabaya Menggunakan Aplikasi IoT Berbasis Jaringan Internet *Wifi*” agar setiap mesin aman dan terkontrol.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah belum adanya pengukuran sensor *accelerometer* dan *proximity* yang dapat dibaca dan dimonitoring dengan menggunakan *Internet of Things* yang terhubung ke internet melalui *wifi*. Di tahun 1991, PT PAL sudah menerapkan jaringan LAN melalui kabel sebagai penghubung mesin-mesin produksi yang dimonitoring melalui akses point. Oleh karena itu, saya menulis Tugas Akhir ini untuk merancang sistem monitoring mesin-mesin dengan menggunakan *Internet of Things* yang terhubung ke internet melalui *wifi* agar dapat mengefisiensi pekerjaan di PT PAL Surabaya.

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, diberikan batasan permasalahan agar pembahasan tidak meluas dan menyimpang dari tujuan. Pada Tugas Akhir ini menggunakan Arduino Uno dan media komunikasi menggunakan ESP8266 *Wifi Shield* agar dapat terhubung dengan jaringan internet menggunakan *router wifi* dan batasan masalah lainnya sebagai berikut :

1. Tugas Akhir ini membahas kecepatan jangkauan pengiriman data menggunakan jaringan internet.
2. Tugas Akhir ini membahas perbandingan pengiriman data sensor pada tampilan *ThingSpeak* dan tampilan di LCD menggunakan *stopwatch* untuk menghitung kecepatan pengiriman data.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang dan membuat sistem monitoring untuk pengefisiensian pekerjaan di industri, diantaranya adalah :

1. Merancang dan membuat sistem monitoring menggunakan aplikasi IoT berbasis jaringan internet *wifi* untuk pengefisiensian pekerjaan di PT PAL Surabaya.
2. Memudahkan perusahaan dalam mengakses data yang dimonitoring dengan menggunakan aplikasi IoT berbasis jaringan internet yang dilakukan oleh pihak perusahaan di PT PAL Surabaya.

1.5 Metodologi Penelitian

Dalam pelaksanaan Tugas Akhir yang berupa monitoring sensor mesin industri di Area Divisi Kapal Niaga PT PAL Surabaya

menggunakan aplikasi IoT berbasis jaringan internet *wifi*, ada beberapa kegiatan yang dapat diuraikan sebagai berikut :

1. Studi Pustaka dan Survey Data Awal

Berikutnya Studi pustaka yang dilakukan dalam hal ini adalah dengan mencari literatur di ruang baca, perpustakaan atau di internet serta untuk pengambilan data awal dilakukannya *survey* lapangan ke Divisi Kapal Niaga PT PAL Surabaya mengenai cara kerja sensor yang terdapat di mesin, mengenai pemrograman arduino, pengalaman pada *web*, pembuatan aplikasi *ThingSpeak* dan tampilan *android* serta mengenai teknologi IoT.

2. Perencanaan Pembuatan Alat

Tahapan ini dilakukan setelah mendapat informasi dari referensi di atas. Perencanaan alat meliputi perancangan rangkaian alat untuk pembacaan sensor oleh arduino yang dapat terkoneksi dengan menggunakan *wifi* dan menampilkan menu monitoring berupa pembacaan data sensor yang tampil di laptop dan router sebelum terkoneksi ke internet. Perancangan perangkat komunikasi untuk tampilan menu monitoring pada aplikasi *android* dan *Personal Computer Home* menggunakan *Internet of Things* yaitu mengoneksikan ke internet menggunakan setting *wifi* yang tersambung ke laptop dan router sebagai penyambung ke internet. Parameter tersebut antar lain :

1. Rancangan sistem monitoring sensor mesin industri menggunakan aplikasi *Internet of Things* berbasis jaringan internet.
2. Desain Alat
3. Bahan yang digunakan
4. Lama waktu pembuatan

3. Perencanaan Pembuatan Software

Agar hasil rancangan dan pembuatan alat bisa bekerja dengan normal, maka perlu pemrograman pada arduino untuk mengkoordinasikan kerja masing-masing komponen. Pembuatan *software* ini dilakukan dengan menggunakan modul ESP8266 yang dapat mengirim aplikasi IoT berupa tampilan *ThingSpeak* dengan menampilkan menu monitoring dibagian tampilan grafik data, sedang untuk sistem komunikasi

menggunakan *router* sebagai *wifi* agar dapat mengakses data pada *ThingSpeak*.

4. Penyusunan Laporan

Penyusunan laporan dilakukan setelah mendapat beberapa data pengujian monitoring sensor dan yang telah tersedia di Area Divisi Kapal Niaga PT PAL Surabaya dengan mengoneksikan ke internet menggunakan tampilan *ThingSpeak* dan dapat diproses melalui aplikasi *android* dan *Personal Computer Home* sudah mencukupi.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Berisi penjelasan dasar teori mengenai konsep yang digunakan dalam memonitoring sensor mesin industri menggunakan aplikasi IoT berbasis jaringan internet. Materi meliputi alat *hardware* serta *software*. *Hardware* antara lain mikrokontroler arduino ESP8266, sensor *proximity* induktif dan sensor *accelerometer*. Untuk tampilan menggunakan *web online* dengan mengakses *software ThingSpeak* sebagai akses *login* dan *display* monitoring sensor dan dapat diakses melalui *handphone android*.

Bab III Perancangan Sistem

Bab ini membahas secara detail tentang monitoring sensor mesin industri menggunakan aplikasi IoT dengan komponen pendukungnya, serta membuat tampilan pada *web online* untuk menampilkan monitoring pada mesin industri.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat data-data pengujian alat pada *hardware* dan *software* secara keseluruhan beserta analisisnya. Data yang di ukur yaitu berupa data

keluaran yang terbaca pada sensor *proximity* dan sensor *accelerometer*. Pengujian yang di lakukan antara lain pengujian alat, *software* dan komunikasi.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Relevansi dari alat ini dapat dimanfaatkan sebagai solusi :

1. Untuk mengetahui hasil monitoring yang telah tersedia
2. Membantu efisiensi pekerja dalam mengawasi mesin produksi
3. Mencegah terjadinya kecurangan pihak pabrik industri pada saat tidak dilakukannya monitoring karena hasil dari pembacaan sensor yang sesuai dan relevan dengan keadaan sebenarnya.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

2.1 Tinjauan Pustaka

Tahun 90-an di PT PAL Surabaya telah melakukan realisasi sistem monitoring mesin-mesin dengan menggunakan kabel *ethernet* yang terhubung ke akses point sebagai pengontrol mesin-mesin. Pada saat itu, PT PAL Surabaya menggunakan kabel *ethernet* sebagai saluran media komunikasi. Hasil yang dicapai terdapat kekurangan yaitu karena banyaknya kabel yang tersambung pada mesin-mesin membuat terlihat kurang nyaman, serta dengan menggunakan kabel *ethernet* memiliki kecanggihan sangat terbatas dan data yang dikirimkan tidak dapat diakses di luar Area PT PAL Surabaya. Oleh karena itu memerlukan adanya inovasi dengan media komunikasi *wifi*.

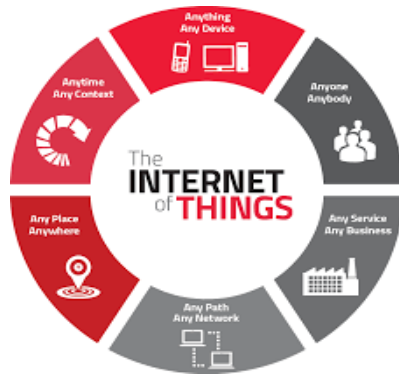
2.2 *Internet of Things*[1]

Internet of Things adalah adalah suatu konsep dimana konektifitas internet dapat bertukar informasi satu sama lainnya dengan benda-benda yang ada disekelilingnya. Banyak yang memprediksi bahwa *Internet of Things* (IoT) merupakan “*the next big thing*” di dunia teknologi informasi. Hal ini dikarenakan banyak sekali potensi yang bisa dikembangkan dengan teknologi *Internet of Things* (IoT) tersebut.

Konsep IoT ini sebetulnya cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yakni: Barang Fisik yang dilengkapi modul IoT, perangkat koneksi ke internet seperti modem dan *router wireless speedy* seperti di rumah anda, dan *cloud data center* tempat untuk menyimpan aplikasi beserta data base.

Penerapan IoT kedepan akan lebih terlihat kepada aspek seperti *smarthome*, atau alat pemantau tubuh manusia. Berbagai perangkat yang terkoneksi tersebut tidak butuh layar dan *keyboard* untuk beroperasi. Tidak perlu pula dicolok ke komputer *desktop* untuk mengunduh datanya. Sebab, *smartphone* dan *tablet* akan menjadi *hub* atau pusat untuk mengontrol berbagai perangkat tersebut.

Gambar 2.1 dibawah ini menggambarkan cakupan dari konsep *Internet of Things* dimana konsep ini menawarkan kemudahan dan kenyamanan dalam mengakses perangkat kita dari mana saja dan kapan saja.



Gambar 2.1 Cakupan *Internet of Things*

2.2.1 Aplikasi *Server Internet of Things*[2]

Server merupakan sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar dan juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, contoh seperti halnya berkas atau data lainnya, dan memberikan akses kepada pengguna. *Server Internet of Things* dalam Tugas Akhir ini menggunakan aplikasi *server open source* yang dikembangkan oleh *Mathworks* yaitu *ThingSpeak server*. Dengan *ThingSpeak* kita dapat melakukan 3 hal yang sangat berguna untuk menampung dan menganalisis data, diantaranya :

a. *Collect*

Mengirimkan/ mengonlinekan data-data sensor ke internet (*ThingSpeak Server*). *ThingSpeak* menyediakan *API Key* untuk mengakses semua fitur yang disediakan. Sebelum dapat melakukan pengumpulan data, terlebih dahulu kita harus punya akun dan *channel* di *ThingSpeak*, akun digunakan untuk *login* ke *ThingSpeak*, *Channel* fungsinya sama halnya tabel pada *MySQL*. Setelah membuat akun dan *channel* kita akan diberikan *Write API Key* dan *Read API Key* yang berguna sebagai identitas saat mengirimkan data ke *Thingspeak* atau membaca data dari *Thingspeak*. Semua komunikasi dengan *Thingspeak* dilakukan dengan *HTTP Method*.

b. *Analyze*

Menganalisa dan memvisualisasikan data. Dengan menyimpan data di *online storage* akan memudahkan dalam mengakses data, *Thingspeak* menyediakan *tools* untuk melakukan analisis dan visualisasi data secara *online*. *Thingspeak* juga *support* dengan program MATLAB. Program MATLAB dapat dijalankan di *server Thingspeak* dan juga bisa dijalankan di *dekstop PC*. Data hasil analisa dapat disajikan dalam bentuk plot, grafik atau *gauge*.

c. *ACT*

Mentrigger suatu perintah/ kegiatan berdasarkan data hasil analisa. Dengan fitur ACT kita dapat mentrigger suatu aksi apabila suatu kondisi hal analisa terpenuhi. Misal, saat temperatur mencapai $> 30^{\circ}$ maka akan mentrigger agar kipas menyala. Untuk bias melakukannya terlebih dahulu harus mengkonfigurasi *TalkBackAPP* di *Thingspeak*.

2.3 *Web*[3]

Web server adalah sebuah aplikasi *server* yang melayani permintaan HTTP atau HTTPS dari *browser* dan mengirimkannya kembali dalam bentuk halaman-halaman web. Halaman-halaman web yang dikirim oleh *web server* biasanya berupa file-file HTML dan CSS yang nantinya akan diparsing atau ditata oleh *browser* sehingga menjadi halaman-halaman web yang bagus dan mudah dibaca.

2.4 *Aplikasi Android*[3]

Aplikasi *Android* adalah sebuah kumpulan perangkat lunak untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi utama *mobile*. *Android* sendiri adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux dan mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka. Awalnya, Google Inc. membeli *Android Inc.* yang merupakan pendatang baru yang membuat *software* untuk ponsel/ *smartphone*. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handset Alliance*, *consorsium* dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk google, HTC, Intel, Motorola, Qualcomm, *E-Mobile* dan Nidia.

2.5 MIT APP *Inventor*[4]

App *Inventor* adalah sebuah *tool* untuk membuat aplikasi *android*, yang menyenangkan dari *tool* ini adalah karena berbasis visual blok *programming*, jadi kita bisa membuat aplikasi tanpa kode satupun. Disebut sebagai visual blok *programming* karena kita akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi *event handler* tertentu dalam membuat aplikasi, dan secara sederhana kita bisa menyebutnya tanpa menuliskan kode program *coding less*.

2.6 Arduino[4]

Arduino adalah pengendali *mikro single board* yang bersifat *open source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang *hardware*-nya memiliki prosesor Atmel AVR dan *software*-nya memiliki bahasa pemrograman sendiri. Pada bagian *software* arduino yang akan digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino.

IDE arduino adalah *software* yang sangat canggih dapat ditulis dengan menggunakan Java. IDE arduino terdiri dari :

1. *Editor Program* yaitu sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler* yaitu sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing* yang bisa dipahami oleh mikrokontroler adalah kode biner itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader* yaitu sebuah modul yang membuat kode biner dari komputer ke dalam memori di dalam papan arduino.

2.6.1 Spesifikasi Arduino UNO R3[4]

Arduino Uno R3 adalah papan pengembangan (*development board*) mikrokontroler yang berbasis chip ATmega328P. Disebut sebagai papan pengembangan karena *board* ini memang berfungsi sebagai arena *prototyping* sirkuit mikrokontroler. Dengan menggunakan papan pengembangan, akan lebih mudah merangkai rangkaian elektronika mikrokontroler dibanding jika memulai merakit ATmega328 dari awal di *breadboard*. Modul Arduino Uno R3 seperti pada Gambar 2.2, Spesifikasi Arduino UNO R3 dapat dilihat pada Tabel 2.1.



Gambar 2.2 Arduino UNO R3

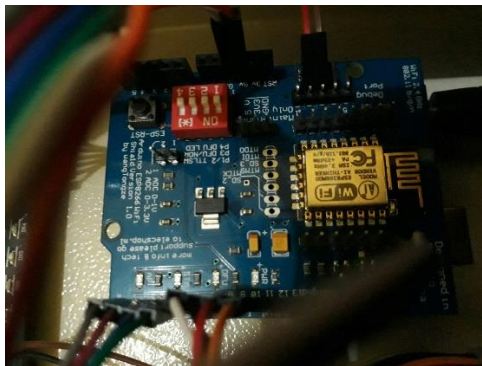
Table 2.1 Spesifikasi Arduino UNO R3

<i>Items</i>	<i>Description</i>
Chip mikrokontroller	ATmega328P
Tegangan operasi	5V
Tegangan input (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	6 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori Flash	32 KB, 0.5 KB telah digunakan untuk bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 Mhz

2.7 ESP8266 Wifi Shield Tipe 12E[5]

ESP8266 *Wifi Shield* dikembangkan oleh *Shenzhen Doctors of Intelligence & Technology (SZDOIT)*, yang pin – pinnya dirancang kompatibel penuh dengan Arduino Uno dan Arduino Due, jadi tinggal pasang saja bak lego ke Arduino Uno. Selain menambahkan kapabilitas arduino untuk terhubung dengan akses point *wifi* di suatu tempat (sebagai *client*), *shield* ini juga menyediakan akses point *wifi* sendiri (*server*), hal inilah yang menjadi salah satu kelebihan dari ESP8266 *wifi shield* dibandingkan dengan *wifi shield* yang lain. Berbeda dengan ESP8266 biasa, ESP tipe 12E ini memiliki *RF shielding*, serta memiliki lebih banyak pin *out* yang dikeluarkan dari IC utama sehingga memiliki lebih banyak I/O yg dapat di akses. Modul *wifi* baru yang pengoprasiaannya terbilang cukup simpel, karena menggunakan *Interface serial AT Command*. Tegangan VCC yg diberikan adalah 3.3V, bukan 5V. ESP8266 modul ini berfungsi untuk membuat proyek *Internet of Things (IoT)*.

ESP8266 *Wifi Shield* ini tidak memerlukan *logic level converter*, karena di dalam *shield* sudah tertanam pengubah logika tegangan. *Shield* ini secara fungsional juga merupakan *expansion board* dari Arduino Uno sekaligus juga *expansion board* ESP8266. Terdapat header pin Arduino Uno dan ESP8266 yang telah disediakan jika kita ingin bereksplorasi dengan port – port yang disediakan. Modul ESP8266 *Wifi Shield* dapat dilihat pada Gambar 2.3, spesifikasi modul ESP8266 *Wifi Shield* tipe 12E dapat dilihat pada Tabel 2.2, dan keterangan port pada ESP8266 *Wifi Shield* dapat dilihat pada Tabel 2.3.



Gambar 2.3 ESP8266 Wifi Shield Tipe 12E

Table 2.2 Spesifikasi ESP8266 Wifi Shield Tipe 12E

<i>Items</i>	<i>Description</i>
<i>Output with resistive Insurance Wire</i>	240V/2A
<i>Voltage version</i>	5 V
<i>Quiescent current</i>	0 Ma
<i>Trigger voltage</i>	0-1.5V
<i>Trigger current</i>	2 Ma
<i>Weight</i>	13g

Table 2.3 Keterangan Port pada ESP8266 Wifi Shield

<i>Port</i>	<i>Pin</i>	<i>Function</i>	<i>Remark</i>
Arduino	G	Ground	
	D	Digital IO	
	A	Analog IO	
	5V	5V	
	TX	Arduino UNO TX	Connection with ESP8266
	RX	Arduino UNO RX	Connection with ESP8266
	SCL	Arduino SCL	
	SDA	Arduino SDA	
ESP 8266 Wifi Shield	G	Ground	
	D	Digital IO	
	3V3	3.3V	
	RX0	ESP8266 RX0	Connection with Arduino
	TX0	ESP8266 TX0	Connection with Arduino
	EN	ESP8266 EN	
	AD	ESP8266 AD	
	RST	ESP8266 reset	
	KEY	ESP8266 Wifi configuration	
	SW	ESP8266 and Arduino	When programming Arduino with IDE, SWI must switch to "OFF" position

2.8 Sensor Proximity Induktif[6]

Sensor *proximity* jenis induktif banyak digunakan untuk mendeteksi adanya benda logam pada jarak tertentu tanpa harus menyentuh benda tersebut. Sensor induktif menggunakan arus induksi oleh medan magnet untuk mendeteksi benda logam di dekatnya. Sensor induktif menggunakan *coil* (induktor) untuk menghasilkan medan magnet frekuensi tinggi seperti yang ditunjukkan pada gambar di bawah ini. Jika ada benda logam di dekat medan magnet yang berubah, arus akan mengalir dalam objek, *shutter* Kamera, Alarm dan lain sebagainya. Bentuk fisik sensor *proximity* induktif dapat dilihat pada Gambar 2.4.

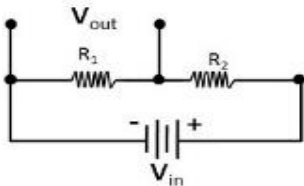


Gambar 2.4 Sensor *Proximity* Induktif

2.9 Perhitungan Perancangan Pembagi Tegangan[6]

Rumus perhitungan dapat dilihat pada Persamaan 2.1, sedangkan gambar rangkaian pembagi tegangan sensor *proximity* induktif dapat dilihat pada Gambar 2.5.

$V_{out}=V_{in} \times (R_1/(R_1+R_2)).....(2.1)$



Gambar 2.5 Rangkaian Pembagi Tegangan untuk Sensor *Proximity* Induktif

2.10 Sensor Accelerometer[7]

Accelerometer adalah sensor yang digunakan oleh sistem untuk mendeteksi orientasi suatu perangkat berdasarkan gerakan ke segala arah atau dengan menggoyangkan yang memungkinkan fitur untuk bertindak. Sesuai namanya *accelerometer* atau akselerasi ini mengukur percepatan bahwa perangkat mengalami perubahan yang relatif sesuai dengan tiga sumbu XYZ atau kanan, kiri, atas, bawah, dan datar. Sensor accelerometer dapat dilihat pada Gambar 2.6.



Gambar 2.6 Sensor Accelerometer

2.11 Lampu Indikator[8]

Lampu indikator panel listrik digunakan sebagai penanda kondisi tertentu pada panel. Pada umumnya terdapat 3 jenis warna yang sering digunakan yakni merah, kuning, dan hijau. Lampu panel listrik teknologi LED memiliki keuntungan yakni tidak panas dan tidak mengkonsumsi daya besar. Lampu Indikator Panel Listrik dapat dilihat pada Gambar 2.7 dan Tabel Spesifikasi Lampu Indikator Panel Listrik dapat dilihat pada Tabel 2.4.



Gambar 2.7 Lampu Indikator

Table 2.4 Spesifikasi Lampu Indikator

<i>Model</i>	AD16-22DS
<i>Operating Voltage</i>	220 VAC
<i>Color</i>	Green, Orange, Red
<i>Luminiscent material</i>	High brightness LED
<i>Lamp head diameter</i>	30mm
<i>Installationhole diameter</i>	22mm

2.12 Relay Module[9]

Modul *relay* ini adalah modul yang sangat praktis untuk digunakan sebagai *main switch relay* dari proyek rangkaian elektronika berbasis mikrokontroler seperti dari *Arduino Development Board* untuk menyalakan/mematikan peralatan elektronika lainnya yang ditenagai listrik AC hingga 250 VAC (listrik PLN) atau perangkat DC bertegangan tinggi (hingga 28 VDC) seperti *High Power DC* motor dengan arus maksimum sebesar 7 *Ampere* (untuk perangkat dengan listrik PLN setara dengan *power rating* ± 1500 Watt). Berikut dibawah ini adalah gambar *relay module* seperti pada Gambar 2.8.



Gambar 2.8 Relay Module

Description:

- *5V 4-Channel Relay interface board*
- *Optical isolation on each channel*
- *Drive directly from Arduino etc.*
- *High power relays for each channel: 250V 10A AC*
- *Indication LED for each output*
- *Module size: 70 x 53 x 20mm*

2.13 LCD (*Liquid Crystal Display*)[10]

LCD (*Liquid Crystal Display*) bisa memunculkan gambar atau tulisan dikarenakan terdapat banyak sekali titik cahaya yang terdiri dari satu buah kristal cair sebagai sebuah titik cahaya. Fungsi pin yang terdapat pada LCD terlihat pada Gambar 2.9. Modul LCD memiliki karakteristik sebagai berikut :

1. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan
2. Terdapat 192 macam karakter
3. Terdapat 80 x 8 *bit display* RAM (maksimal 80 karakter)
4. Memiliki kemampuan penulisan dengan 8 *bit* maupun dengan 4 *bit*
5. Dibangun dengan *osilator local*
6. Satu sumber tegangan 5 Volt
7. Otomatis *reset* saat tegangan dihidupkan
8. Bekerja pada suhu 0°C sampai 55°C



Gambar 2.9 LCD

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

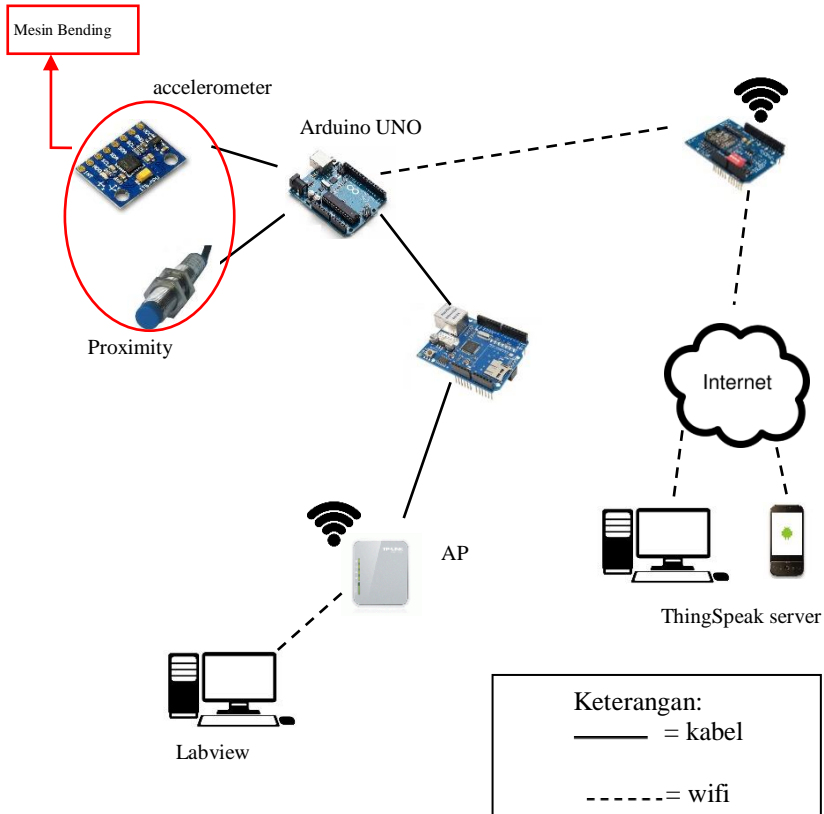
Dalam bab ini dibahas perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Hal tersebut guna mewujudkan Tugas Akhir yang berjudul “Monitoring Sensor Mesin Industri Di Area Divisi Kapal Niaga PT PAL Surabaya Menggunakan Aplikasi IoT Berbasis Jaringan Internet *Wifi*”. Perancangan alat dibahas perbagian disertai dengan gambar skematik. Sedangkan penjelasan *software* akan dijelaskan mengenai pembuatan program *server* dengan menggunakan *IP* pada *website*, pembuatan program *client* dengan menggunakan program Arduino Uno R3 yang dilengkapi dengan *Wifi Shield* ESP8266 Tipe 12E dan *wifi shield* sebagai penghubung jaringan internet *wifi* dan menggunakan *handphone android* untuk tampilan monitoring sensor agar dapat saling terkoneksi dengan baik.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi dua yaitu:

- a. Perancangan *hardware* (perangkat keras) yang terdiri dari pembuatan rangkaian untuk pembacaan kedua sensor, perancangan untuk modul relay sebagai indikator, serta perancangan hardware dari sistem keseluruhan.
- b. Perancangan *software* (perangkat lunak) meliputi perancangan pembuatan program *server* dengan menggunakan *Thingspeak* untuk aplikasi IoT, dan *App Invertor* untuk aplikasi di *android*.

3.1 Diagram Blok Skenario Kerja Alat

Sistem ini bekerja sesuai dengan pembacaan sensor di mesin industri. Dimana dalam PT PAL Surabaya telah terpasang *router wifi* sebagai media komunikasi jarak jauh. *Arduino* dikoneksikan melalui ESP8266 *wifi*, dengan demikian pembacaan sensor di mesin industri menggunakan aplikasi IoT dapat terhubung dengan *wifi* dari jarak jauh. Kondisi dari objek yang dimonitoring dapat dilihat pada *android* dari tempat manapun asalkan masih ada jaringan *server*. Adapun diagram blok desain skenario alat dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem

Dari Gambar 3.1 Diagram Blok Sistem terdapat beberapa komponen yang berperan dalam fungsinya masing-masing. Pada fungsi nyata sensor *accelerometer* sebagai sensor yang terdapat pada suatu mesin industri yaitu mesin bending dalam penentuan besar derajat suatu pembengkokan besi dalam pembuatan besi-besi untuk konstruksi kapal di Area Divisi Kapal Niaga PT PAL Surabaya. Dan untuk sensor *proximity* sebagai sensor untuk pendeteksi ada tidaknya suatu besi yang masuk ke dalam area pembengkokan besi pada mesin bending di Divisi Kapal Niaga PT PAL Surabaya. Untuk memonitoring keadaan kedua sensor tersebut, PT PAL Surabaya harus melihat langsung data sensor pada mesin. Dalam Tugas Akhir ini, penulis membahas 2 masalah yang

diangkat pada PT PAL Surabaya yaitu pertama, membahas kecepatan jangkauan pengiriman data menggunakan jaringan internet. Dan kedua, membahas perbandingan pengiriman data sensor pada tampilan *ThingSpeak* dan tampilan di lcd menggunakan *stopwatch* untuk menghitung kecepatan pengiriman data.

Pada Tugas Akhir ini, penulis mengambil kegunaan dari kedua sensor yang diaplikasikan di PT PAL Surabaya tersebut. Dari kedua sensor tersebut, penulis membuat *prototype* yang berfungsi sebagai media komunikasi dalam memonitoring sensor dari jarak jauh. *Accelerometer* yang digunakan yaitu mengidentifikasi sumbu X dan sumbu Y saat sensor *accelerometer* bekerja, sedang untuk sensor *proximity* mengidentifikasi adanya bimetal bilamana jika teridentifikasi bimetal menunjukkan angka 1 dan bila tidak teridentifikasi bimetal menunjukkan angka 0.

Penulis juga menggunakan komponen elektronika lainnya sebagai media komunikasi agar dapat terhubung dengan jaringan internet sehingga dalam melakukan monitoring data sensor tidak perlu secara langsung melihat ke mesin. Komponen elektronika sebagai media komunikasi yang digunakan diantaranya, arduino uno dan *wifi shield*. Arduino uno yang digunakan dalam *prototype* ini difungsikan sebagai pembacaan input dari sensor yang dihubungkan. Sedang untuk *wifi shield* yang digunakan menggunakan tipe ESP8266 tipe 12E yang berfungsi sebagai media agar sensor yang terbaca dapat terhubung ke internet. Kemudian data sensor dapat diakses melalui *ThingSpeak* agar dapat dimonitoring melalui jarak jauh.

Dibawah ini adalah rumus untuk mencari rangkaian pembagi tegangan yang diberikan untuk sensor *proximity* pada Persamaan 2.1 dibawah ini :

$$V_{out} = V_{in} \times (R_1 / (R_1 + R_2)) \dots \dots \dots (2.1)$$

dengan $V_{out} = 5V$

$$R_1 = 13 \Omega$$

$$R_2 = 18 \Omega$$

Maka R_1 dan R_2 didapatkan hasil :

$$V_{out} = V_{in} \times (R_1 / (R_1 + R_2))$$

$$5V = V_{in} \times (13 \Omega / (13 \Omega + 18 \Omega))$$

$$5V = V_{in} \times 13 \Omega / (31 \Omega)$$

$$V_{in} = 12V$$

Rangkaian pembagi tegangan seperti pada Persamaan 2.1 diatas digunakan untuk mencari hasil dari perhitungan resistor agar dapat V_{in} pada arduino bisa mencukupi 5 Volt. Seperti yang diketahui jika R_1

mempunyai nilai $13\ \Omega$ dan R_2 mempunyai nilai $18\ \Omega$. Maka akan didapatkan hasil V_{in} dari sensor *proximity* yang membutuhkan 12 Volt.

3.2 Implementasi dan Konfigurasi Wifi Shield

Instalasi dan konfigurasi meliputi pemrograman yang dilakukan di arduino dan *wifi shield*. Untuk memprogram *proximity* agar pembacaan data dapat tampil di tampilan aplikasi *android* yaitu dengan menggabungkan program *proximity* dan program koneksi *wifi* oleh ESP8266 *Wifi Shield*.

3.2.1 Proses Konfigurasi Server IoT

Server yang digunakan dalam Tugas Akhir ini memanfaatkan fasilitas yang disediakan oleh *MATHWORKS* berupa *server* IoT yang diberi nama *Thingspeak*, *server* ini bisa diakses di alamat <http://thingspeak.com>. Untuk dapat menggunakan *server* ini terlebih dahulu harus mendaftarkan akun terlebih dahulu. Proses pendaftaran akun sama halnya dengan pendaftaran akun pada *website* umumnya. Setelah mendaftarkan akun berikutnya adalah proses pembuatan *channel* dimana *channel* inilah yang akan menampung dan menampilkan data-data sensor yang diunggah.

3.2.1.1 Pendaftaran Akun ThingSpeak

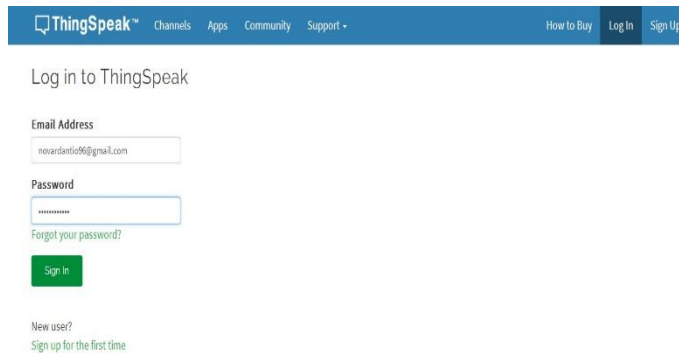
Pada tampilan ini pada Gambar 3.2, terlebih dahulu sebelum *login* ke *Thingspeak* kita diwajibkan untuk membuat akun dengan mensinkronisasikan alamat *email* serta memasukkan *password* yang sesuai. Terdapat petunjuk untuk membuat akun *Thingspeak* dengan sangat mudah.



Gambar 3.2 Sign Up ThingSpeak

3.2.1.2 Login Akun *ThingSpeak*

Setelah membuat akun *Thingspeak*, untuk membuka halaman *Thingspeak* terlihat pada Gambar 3.3 Tampilan *login Thingspeak* dengan memasukkan alamat *email* dan *password* yang telah dibuat.

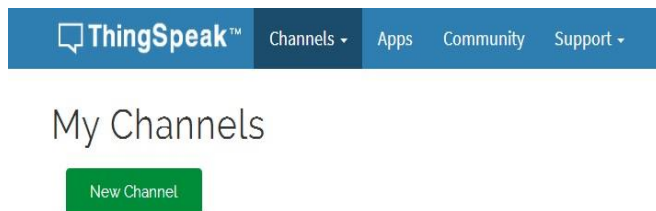


Gambar 3.3 *Login Thingspeak*

3.2.1.3 Konfigurasi *Channel*

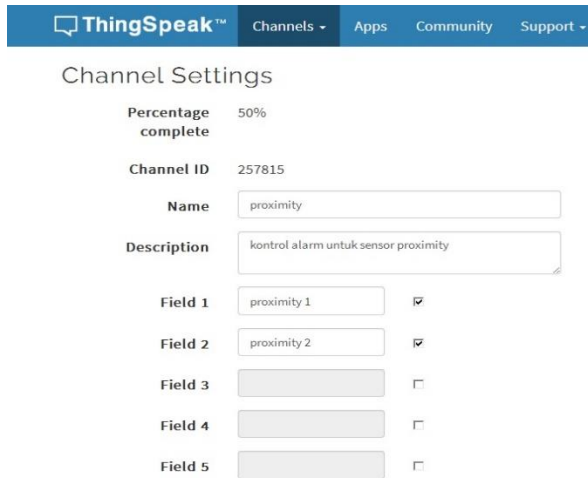
Channel merupakan tabel data yang nantinya akan digunakan menyimpan data yang diunggah. Berikut adalah proses pembuatan *channel* dan konfigurasinya :

- a. Pada menu “*My Channel*” klik “*New Channel*” seperti pada Gambar 3.4 berikut ini :



Gambar 3.4 *Membuat Channel di ThingSpeak*

- b. Kemudian akan dihadapkan dengan “*Channel Setting*”, pada tahapan ini perlu diisi *field* apa saja yang akan digunakan, misalnya seperti Gambar 3.5 berikut:



ThingSpeak™ Channels Apps Community Support

Channel Settings

Percentage complete 50%

Channel ID 257815

Name proximity

Description kontrol alarm untuk sensor proximity

Field 1 proximity 1 ☒

Field 2 proximity 2 ☒

Field 3 ☐

Field 4 ☐

Field 5 ☐

Gambar 3.5 Konfigurasi *Channel* di *Thingspeak*

- c. Setelah itu klik pada “*API Keys*”, di bagian ini akan diperoleh *API Key* yang diperlukan agar kita dapat mengakses *channel* yang berada di *server* dengan menggunakan *device* kita. Gambar 3.6 menunjukkan *API Key* yang didapatkan :



ThingSpeak™ Channels Apps Community Support

Private View Public View Channel Settings API Keys Data Import / I

Write API Key

Key O1HRE8DIYYI4L5O3

Generate New Write API Key

Read API Keys

Key LUCUSIITHEUDJWLS

Note

Save Note Delete API Key

Gambar 3.6 *API Key*

3.3 Implementasi dan Konfigurasi Aplikasi *Android*

Aplikasi *android* pada Tugas Akhir ini digunakan sebagai antar muka ke pengguna untuk memonitoring pembacaan data sensor yang telah terukur. Aplikasi ini dibuat dengan tujuan memiliki antar muka yang sederhana dan dapat dioperasikan dengan mudah. Aplikasi *android* yang digunakan pada Tugas Akhir ini yaitu menggunakan App *Inventor* yang mencakup tampilan informasi sensor baik berupa dalam bentuk grafik dari *ThingSpeak* yang langsung disambungkan dengan alamat *channel* ID yang telah tersedia dari kedua sensor yang dimonitoring.

3.3.1 Desain pada APP *Inventor*

Sebelum pengaplikasian tampilan pada *handphone android*, terlebih dahulu harus membuat desain App *Inventor* dengan *login email* pada alamat <http://ai2.appinventor.mit.edu/>, maka akan keluar tampilan untuk membuat project baru.

Aplikasi *android* yang digunakan pada *handphone* menggunakan *software* APP *Inventor*. Berikut tampilan *login* untuk telemetering sensor mesin industri pada *handphone android* :

- a. Tampilan *Screen* 1 terlihat pada Gambar 3.7 berikut :



Gambar 3.7 Membuat Desain Tampilan *Screen* 1 pada MIT App *Inventor*

Dengan memasukkan *password* : tugasakhir

b. Tampilan *Screen 2* terlihat pada Gambar 3.8 berikut :

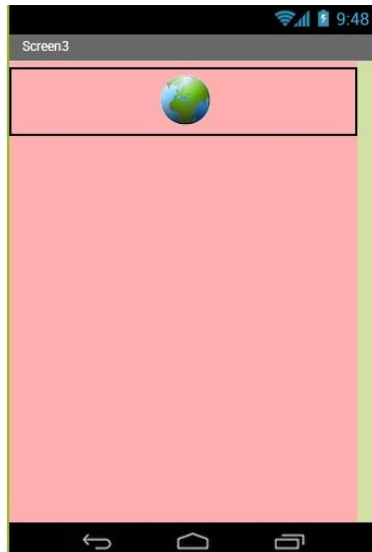


Gambar 3.8 Membuat Desain Tampilan *Screen 2* pada MIT App Inventor

Terdapat 3 pilihan menu yaitu :

1. *Go To URL Proximity* adalah untuk menampilkan telemetering sensor *proximity* yang langsung dapat ditampilkan di *screen 3* yaitu alamat web beserta id *channel* telemetering *proximity* pada *field chart ThingSpeak*.
Channel id : 257815
2. *Go To URL Accelerometer* adalah untuk menampilkan telemetering sensor *accelerometer* yang langsung dapat ditampilkan di *screen 4* yaitu alamat web beserta id *channel* telemetering *accelerometer* pada *field chart ThingSpeak*.
Channel id : 257820
3. *Go To MAP* adalah untuk menampilkan petunjuk arah jalan/ *maps* di Area Divisi Kapal Niaga PT PAL jika di klik langsung menampilkan petunjuk arah jalan di *screen 5*.
4. *Logout* adalah untuk mengembalikan ke halaman awal monitoring, tepatnya pada tampilan *login* telemetering sensor.

c. Tampilan *Screen 3*, 4 dan 5 terlihat pada Gambar 3.9, berikut :

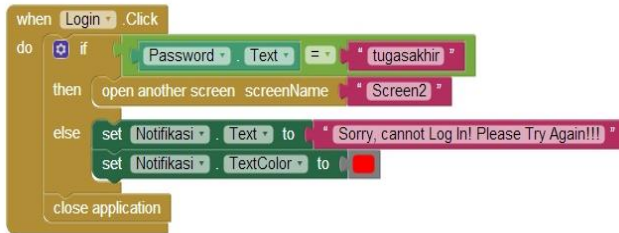


Gambar 3.9 Membuat Desain Tampilan *Screen 3*, 4 dan 5 pada MIT App Inventor

Pada Gambar 3.9 diatas menggunakan *interface web viewer* yang menghubungkan ke internet untuk dapat mengakses aplikasi *ThingSpeak*. Pada desain tampilan blok *screen 3*, ketika ingin menampilkan data sensor *proximity* yang ingin dimonitoring. *Web viewer* ini akan *mlink*-kan satu *channel* dan satu *fieldchart* data sensor *proximity*. Sedangkan untuk tampilan blok *screen 4*, *web viewer* akan *mlink*-kan data sensor *accelerometer* pada satu *channel* dua *fieldchart* yaitu *fieldchart* pertama berupa sumbu X dan *fieldchart* kedua berupa sumbu Y dari hasil data sensor *accelerometer*. Dan tampilan untuk tampilan blok *screen 5* dapat *mlink*-kan ke akses denah lokasi penerapan monitoring di PT PAL Surabaya.

3.3.2 Membuat Tampilan *Login* Blok pada APP *Invertor*

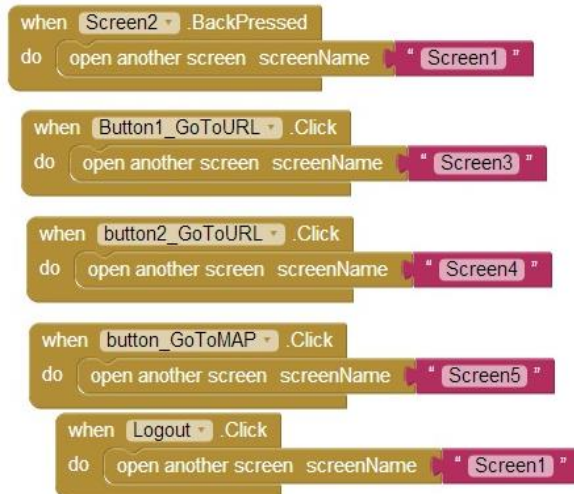
a. Tampilan *Screen 1* terlihat pada Gambar 3.10 berikut :



Gambar 3.10 Membuat Blok Tampilan *Screen 1* pada MIT APP *Invertor*

Pada Gambar 3.10 blok tampilan *screen 1* APP *Invertor* diatas menunjukkan bahwa ketika pengguna memasukkan *password* dengan nama “tugasakhir”, maka pada *screen 1* akan membuka tampilan pada *screen 2*.

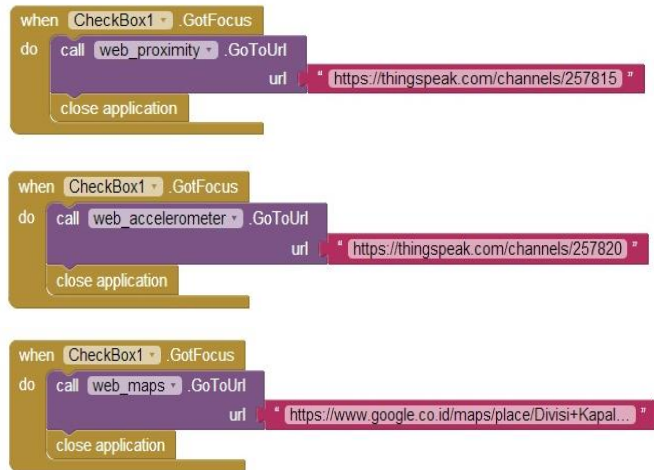
b. Tampilan *Screen 2* terlihat pada Gambar 3.11 berikut :



Gambar 3.11 Membuat Blok Tampilan *Screen 2* pada MIT APP *Invertor*

Dari Gambar 3.11 diatas menjelaskan bahwa pada tampilan *Screen* 2, terdapat 3 pilihan yang dapat menghubungkan ke layanan monitoring sensor *proximity*, sensor *accelerometer*, dan peta ke PT PAL Surabaya. Dari ketiga pilihan ini masing-masing menghubungkan ke beberapa *screen* sesuai dengan blok yang telah dibuat.

c. Tampilan *Screen* 3, 4 dan 5 terlihat pada Gambar 3.12 berikut :

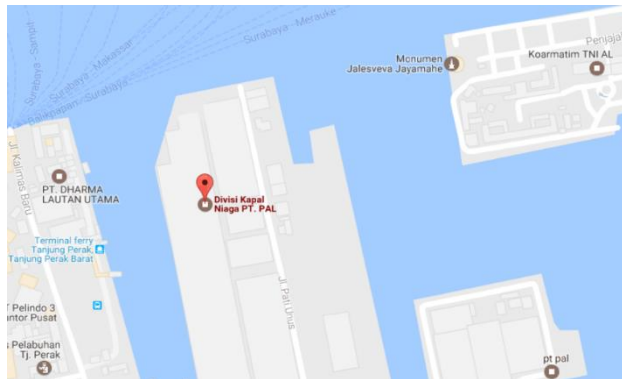


Gambar 3.12 Membuat Blok Tampilan *Screen* 3,4 dan 5 pada MIT APP Invertor

Dari Gambar 3.12 diatas adalah tampilan untuk menampilkan monitoring sensor *proximity* pada aplikasi *ThingSpeak* yang terhubung dengan internet dengan memasukkan alamat url *ThingSpeak* beserta ID *channel* tampilan sensor *proximity* pada *ThingSpeak*. Serta gambar blok lainnya adalah tampilan untuk menampilkan monitoring sensor *accelerometer* pada aplikasi *ThingSpeak* yang terhubung dengan internet dengan memasukkan alamat url *ThingSpeak* beserta id *channel* tampilan sensor *accelerometer* pada *ThingSpeak*. Dan gambar yang terakhir adalah tampilan untuk menampilkan peta PT PAL Surabaya yang terhubung ke internet dengan memasukkan alamat url PT PAL Surabaya.

3.4 Lokasi Penerapan

PT PAL Surabaya Divisi Kapal Niaga beralamat di Jl. Pati Unus, Ujung, Semampir, Kota SBY, Jawa Timur 60155. Jarak penerapan untuk pemasangan jaringan telemetering sekitar kurang lebih 500 meter (terhitung dari Akses point area hingga ke mesin). Lokasi Penerapan yang berada di PT PAL Surabaya dapat dilihat pada Gambar 3.13 dan untuk denah lokasi pemasangan alat monitoring yang berada di PT PAL Surabaya dapat dilihat pada Gambar 3.14.



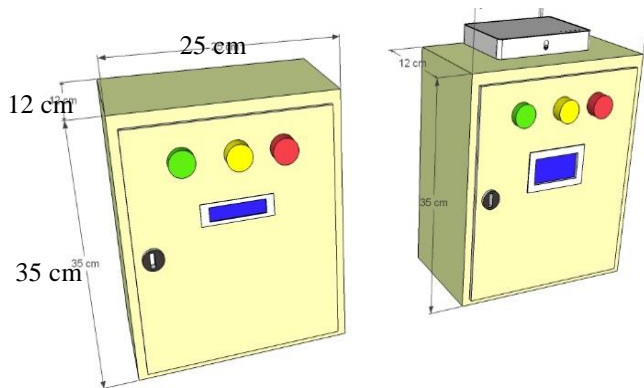
Gambar 3.13 Lokasi Penerapan Jaringan *Wifi* dan *Ethernet* di PT PAL Surabaya



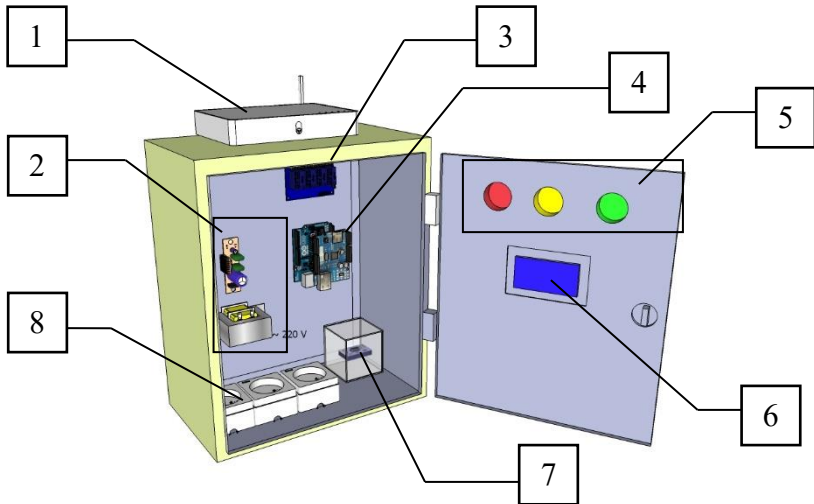
Gambar 3.14 Denah Lokasi Jaringan *Wifi* dan *Ethernet* di PT PAL Surabaya

3.5 Perancangan *Hardware*

Untuk perancangan *hardware* menggunakan panel kontrol, yang didalamnya terdapat Arduino Uno R3, ESP8266 *Wifi shield* tipe 12E, *Ethernet* W5100, alarm kontrol, lampu indikator, LCD, dll. Dimana pada pintu panel dipasang lampu indikator dan LCD. Perancangan pintu dapat dilihat pada Gambar 3.15 dengan ukuran 2 x 30 cm. Rancangan posisi alat pada mekanik kotak panel didalamnya terdapat komponen arduino UNO R3, ESP8266 *Wifi shield*, sensor *accelerometer*, sensor *proximity*, supply dan alarm kontrol. Sedang untuk panel yang satunya berisi sama dengan panel yang satunya yang telah disebutkan, hanya saja memakai sensor *proximity* yang dapat dilihat pada Gambar 3.15 untuk rancangan bagian pintu panel, Gambar 3.16 untuk rancangan posisi sensor *accelerometer* pada mekanik bagian dalam kotak panel dan Gambar 3.17 untuk rancangan posisi sensor *proximity* pada mekanik bagian dalam kotak panel.



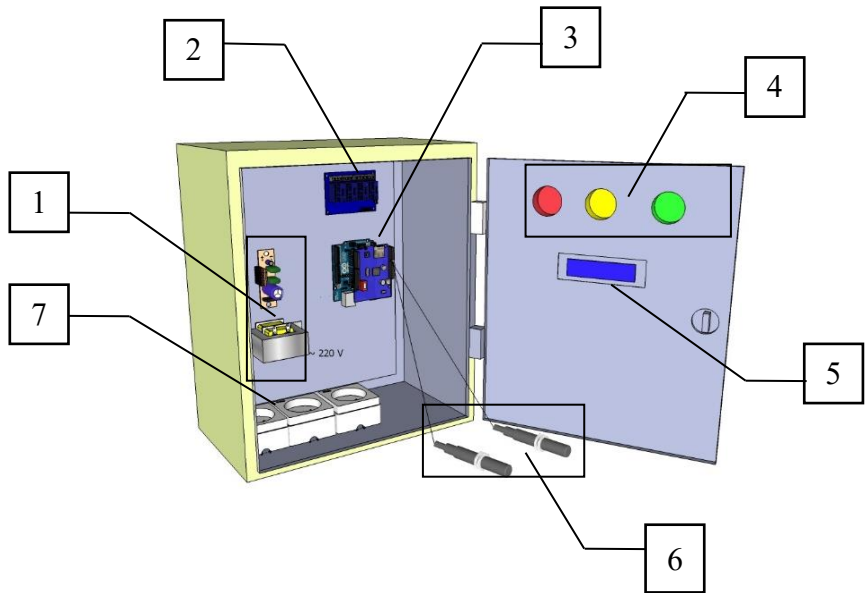
Gambar 3.15 Rancangan Bagian Pintu Panel



Keterangan :

1. *Access Point*
2. *Power Supply*
3. *Relay*
4. *Arduino Uno dan ESP8266 Wifi Shield tipe 12E*
5. *Lampu indikator*
6. *LCD*
7. *Sensor Accelerometer*
8. *Stop Kontak*

Gambar 3.16 Rancangan Posisi Sensor *Accelerometer* Pada Mekanik Bagian Dalam Kotak Panel

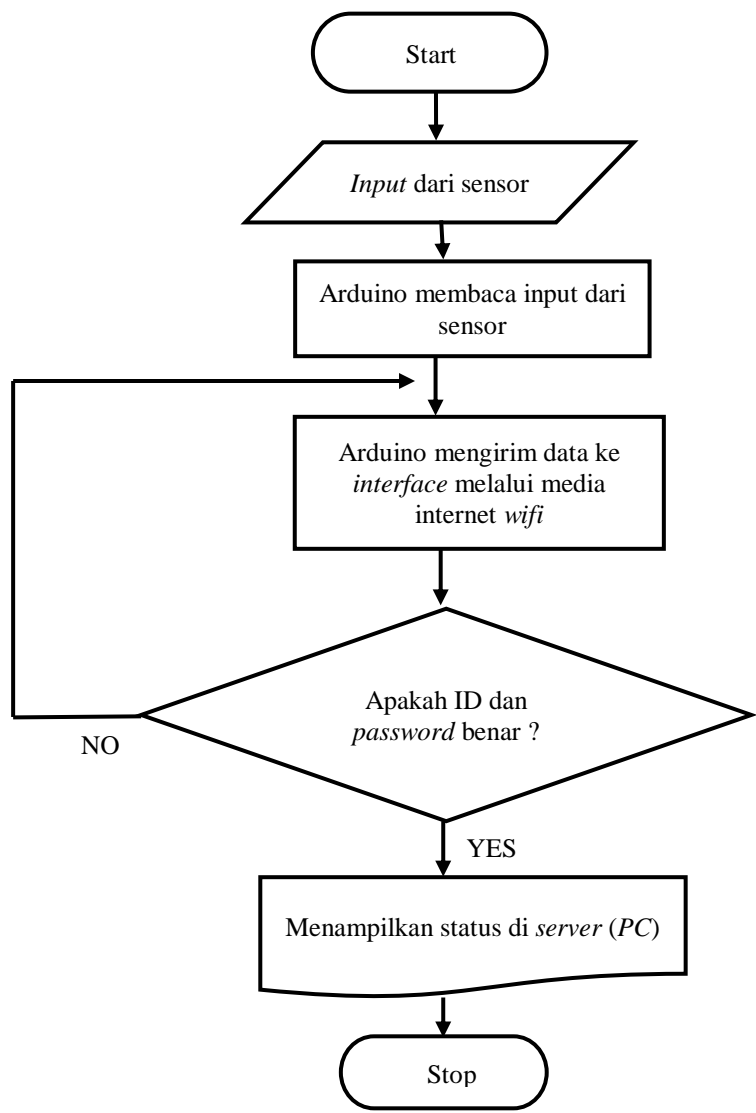


Keterangan :

1. *Power Supply*
2. *Relay*
3. *Arduino Uno dan Ethernet*
4. *Lampu indikator*
5. *LCD*
6. *Sensor Proximity*
7. *Stop Kontak*

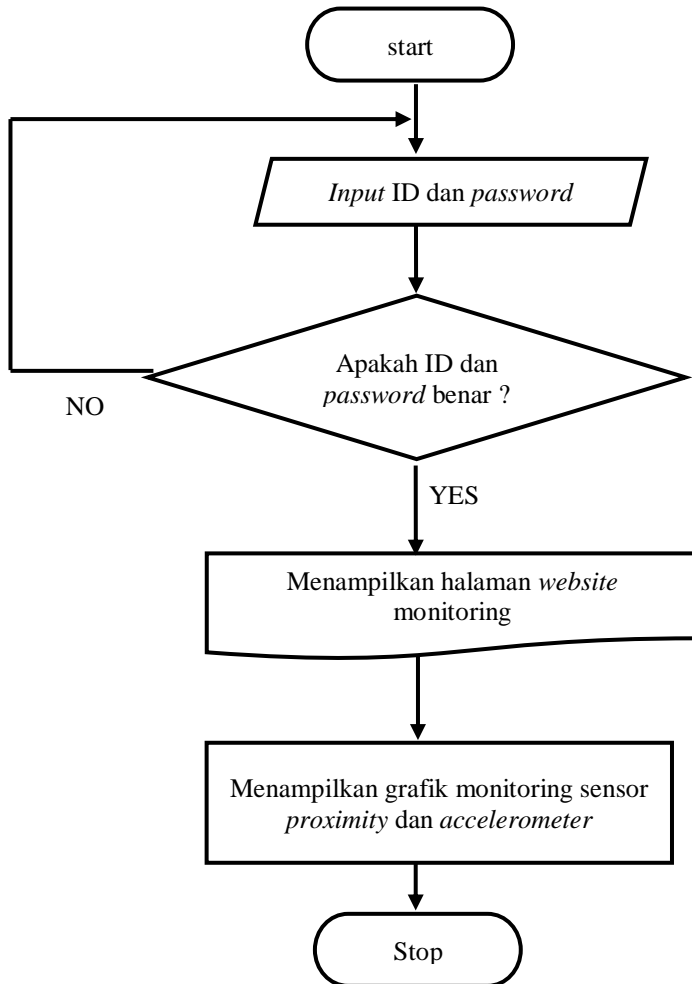
Gambar 3.17 Rancangan Posisi Sensor *Proximity* pada Mekanik Bagian Dalam Kotak Panel

3.6 Perancangan *Software*



Gambar 3.18 *Flowchart Status*

Dari Perancangan *Software* pada Gambar 3.18 *flowchart* status tersebut dapat dijelaskan bahwa *flowchart* status ini menjelaskan, arduino membaca input dari sensor lalu arduino mengirim data sensor ke *interface* (akun *ThingSpeak*) melalui media komunikasi *wifi* (jaringan internet). Jika internet *wifi* dan arduino terhubung maka akan menampilkan status di *personal computer* atau *server* serta *handphone*. Jika internet *wifi* dan arduino tidak terhubung maka arduino yang terhubung dengan *wifi* internet akan mengirim data sensor lagi ke akun *ThingSpeak*. Data yang dikirimkan pada akun *ThingSpeak* berupa tampilan grafik data.



Gambar 3.19 *Flowchart* Pembacaan Data Sensor

Dari perancangan *Software* pada Gambar 3.19 tersebut dapat dijelaskan bahwa *software* akun *ThingSpeak* terbuka untuk umum, ketika sudah mendaftar di akun *ThingSpeak* maka dalam *login* akun *ThingSpeak* ini diperlukan alamat *email* dan *password*. ID alamat *email* yang digunakan akun *ThingSpeak* ini yaitu novardantio96@gmail.com dengan menggunakan *password* surabaya4NOV. Setelah memasukkan alamat *email* dan *password* tersebut, akun *ThingSpeak* terbuka. Tetapi, jika alamat *email* dan *password* belum benar maka akan kembali ke *flowchart* di bagian *login*. Ketika sudah me-*login* maka akun *ThingSpeak* akan menampilkan halaman *website* monitoring. Dalam halaman *website* monitoring terdapat 2 *channel* yaitu *channel* monitoring untuk memonitoring data sensor *accelerometer* dan memonitoring data sensor *proximity*. Dari 2 *channel* tersebut terdapat ID *channel* yang berbeda, yaitu untuk sensor *accelerometer* ID nya 257820 dan untuk sensor *proximity* yaitu 257815. Kemudian untuk me-*logout* akun *ThingSpeak* dapat digunakan pilihan *logout*.

----Halaman ini sengaja dikosongkan----

BAB IV

PENGUJIAN DAN ANALISA

Bab ini dibagi menjadi lima bagian. Pada bagian pertama dijelaskan mengenai hasil pengujian pada media komunikasi yaitu koneksi *wifi shield* yang digunakan dan pada bagian kedua akan dijelaskan mengenai hasil pengujian pada kecepatan pengiriman data sensor dengan menggunakan jaringan internet, sedang pada bagian ketiga pengujian dilakukan pada pengambilan data sensor dengan membandingkan hasil dari sensor menggunakan alat pengukur yaitu busur derajat, keempat adalah hasil pengujian *software* yang digunakan sebagai tampilan pembacaan data sensor secara *realtime*. Dan kelima membahas pengujian keseluruhan dari pengambilan data, untuk gambar hasil pengujian alat terlampir.

Pengujian juga merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem. Pengujian tersebut meliputi :

1. Pengujian Koneksi Wifi Shield
2. Pengujian Kecepatan Pengiriman Data Sensor
3. Pengujian Sensor
4. Pengujian *Software ThingSpeak*
5. Pengujian Keseluruhan

4.1 Pengujian Koneksi Wifi Shield

Pengujian koneksi antara *Personal Computer* dengan *router wifi* dilakukan dengan menggunakan program *Command Prompt*. *Personal Computer* memiliki alamat IP 192.168.8.101, *router wifi* memiliki alamat IP 192.168.8.1, sedangkan untuk *Wifi Shield* arduino diprogram dengan alamat IP 192.168.8.103.

Sebelum melakukan pengujian antara *Personal Computer* dengan *Wifi Shield* arduino, Koneksi antara *Personal Computer* dengan *router wifi* diuji terlebih dahulu yaitu dengan mengetahui IP *address* dari *Personal Computer*. Untuk tampilan pengujian IP *Address Personal Computer* dapat dilihat pada Gambar 4.1, sedang untuk tampilan pengujian koneksi antara *personal computer* dan *esp8266 wifi shield* dapat dilihat pada Gambar 4.2, dan tampilan pengujian kedua koneksi antara *personal computer* dan *router wifi* dapat dilihat pada Gambar 4.3 dibawah ini.

```
Connection-specific DNS Suffix . : Microsoft Wi-Fi Direct Virtu
Physical Address. . . . . : 56-8C-A0-B0-BD-E1
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi 4:

Connection-specific DNS Suffix . :
Description . . . . . : Realtek RTL8723BE 802.11 bgn
Physical Address. . . . . : 54-9C-A0-B0-BD-E1
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::bc26:e978:7aeb:6b4b%3
IPv4 Address. . . . . : 192.168.8.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Thursday, June 8, 2017 4:12:
Lease Expires . . . . . : Friday, June 9, 2017 4:24:50
Default Gateway . . . . . : 192.168.8.1
DHCP Server . . . . . : 192.168.8.1
```

Gambar 4.1 Tampilan Pengujian IP Address *Personal Computer*

Dari Gambar 4.1 diatas menerangkan pada bagian lingkaran merah terlihat bahwa IP address yang digunakan pada *personal computer* memiliki alamat IP 192.168.8.101.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\audy>ping 192.168.8.103

Pinging 192.168.8.103 with 32 bytes of data:
Reply from 192.168.8.101: Destination host unreachable.
Reply from 192.168.8.1: Destination host unreachable.
Reply from 192.168.8.1: Destination host unreachable.
Reply from 192.168.8.1: Destination host unreachable.

Ping statistics for 192.168.8.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\audy>
```

Gambar 4.2 Tampilan Pengujian Koneksi antara *Personal Computer* dan ESP8266 *Wifi Shield*

Dari Gambar 4.2 diatas menerangkan pada bagian lingkaran merah terlihat bahwa ESP8266 *wifi shield* pada *personal computer* memiliki alamat IP 192.168.8.103.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\audy>ping 192.168.8.1

Pinging 192.168.8.1 with 32 bytes of data:
Reply from 192.168.8.1: bytes=32 time=5ms TTL=64
Reply from 192.168.8.1: bytes=32 time=3ms TTL=64
Reply from 192.168.8.1: bytes=32 time=2ms TTL=64
Reply from 192.168.8.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.8.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 5ms, Average = 3ms

C:\Users\audy>_

```

Gambar 4.3 Tampilan Pengujian Kedua Koneksi antara *Personal Computer* dan *Router Wifi*

Dari Gambar 4.3 diatas menerangkan pada bagian lingkaran merah terlihat bahwa *router wifi* pada *personal computer* memiliki alamat IP 192.168.8.1.

4.2 Pengujian Kecepatan Pengiriman Data Sensor di Jaringan Internet *Wifi*

Table 4.1 Pengujian Sensor di Jaringan Internet *Wifi*

Waktu	Sumbu X	Sumbu Y
Delay 15 detik	12	-1
	13	-1
	12	-1
	13	-1
	12	-1
	17	11
	-12	4
	-21	4
	42	-7
	44	-4
Waktu	Sumbu X	Sumbu Y
Delay 20 detik	14	-2
	13	-1
	19	0
	11	0
	-1	3
	0	-2
	0	-9
	6	6
	1	-3
	-1	-1

Waktu	Sumbu X	Sumbu Y
Delay 25 detik	12	0
	13	0
	5	-3
	11	0
	12	0
	17	-3
	5	0
	-4	-1
	2	9
	1	-2

Waktu	Sumbu X	Sumbu Y
Delay 30 detik	12	0
	11	15
	10	12
	8	5
	7	-16
	6	3
	2	-3
	2	-5
	0	-1
	5	1

Pewaktu untuk pengujian pembacaan sensor pada *serial monitor* yang tertulis pada Tabel 4.1 diatas menggunakan *stopwatch handphone*, analisa lamanya pengiriman data adalah sebagai berikut :

1. Dari *delay* 15 detik dengan pengambilan data sensor yang terbaca pada *serial monitor* menggunakan pewaktu *stopwatch* sebagai parameter lamanya pengiriman data yang terbaca di *serial monitor* dan dibandingkan dengan lamanya pengiriman data sensor pada *ThingSpeak*. Untuk hasil lamanya pengiriman menggunakan pewaktu *stopwatch* yang dapat dilihat pada Tabel 4.2.

Table 4.2 Pewaktu Pengiriman Data pada *Serial Monitor*

Hitungan-ke	Detik- ke	Selisih per-detik
01	00.15.27	00.15.27
02	00.40.40	00.25.13
03	01.02.11	00.21.71
04	01.14.45	00.12.34
05	01.24.52	00.10.07
06	01.58.58	00.34.06
07	02.09.63	00.11.05

Hitungan- ke	Detik- ke	Selisih per-detik
08	02.36.45	00.26.82
09	03.00.50	00.24.05
10	03.23.54	00.23.04
11	03.49.07	00.25.53
12	04.03.39	00.14.32
13	04.24.56	00.21.17
14	04.44.02	00.19.46
15	05.01.71	00.17.69
16	05.27.25	00.25.54
17	05.46.66	00.19.41
18	06.12.86	00.26.20

Pembacaan Data pada *ThingSpeak* dengan *delay* 15 detik menunjukkan hasil sebagai berikut :



Gambar 4.4 Hasil Pengiriman Sensor pada *ThingSpeak* dengan *Delay* 15 detik

Pengiriman sensor yang terbaca pada Gambar 4.4 di *ThingSpeak* juga menunjukkan data sensor yang terbaca pada *serial monitor* dengan menunjukkan hasil data yang dibaca adalah sama, hanya saja pengiriman data yang terbaca di *serial monitor* lebih cepat terupload pada *serial monitor* daripada data sensor yang terupload di *ThingSpeak*. Dari perbandingan pernyataan diatas menunjukkan dengan *delay* 15 detik, pada pembacaan *serial monitor* dan pembacaan di *ThingSpeak* hasil data yang terbaca sesuai.

2. Dari *delay* 20 detik dengan pengambilan data sensor yang terbaca pada *serial monitor* menggunakan pewaktu *stopwatch* sebagai

parameter lamanya pengiriman data yang terbaca di *serial monitor* dan dibandingkan dengan lamanya pengiriman data sensor pada *ThingSpeak*. Untuk hasil lamanya pengiriman menggunakan pewaktu *stopwatch* yang dapat dilihat pada Tabel 4.3.

Table 4.3 Pewaktu Pengiriman Data pada *Serial Monitor*

Hitungan- ke	Detik- ke	Selisih per- detik
01	00.11.61	00.11.61
02	00.33.98	00.22.37
03	01.00.08	09.26.10
04	01.27.43	00.27.35
05	01.58.18	00.30.75
06	02.24.28	00.26.10
07	02.53.98	00.29.70
08	03.23.96	00.29.98
09	03.32.10	00.12.70
10	03.50.00	00.17.90
11	04.19.58	00.29.58
12	05.18.23	00.58.65
13	05.41.95	00.23.72
14	06.30.84	00.48.89
15	06.50.78	00.19.94
16	07.01.04	00.10.26
17	07.25.23	00.24.19
18	07.54.40	00.29.17

Pembacaan Data pada *ThingSpeak* dengan *delay* 20 detik menunjukkan sebagai berikut :



Gambar 4.5 Hasil Pengiriman Sensor pada *ThingSpeak* dengan *Delay* 20 detik

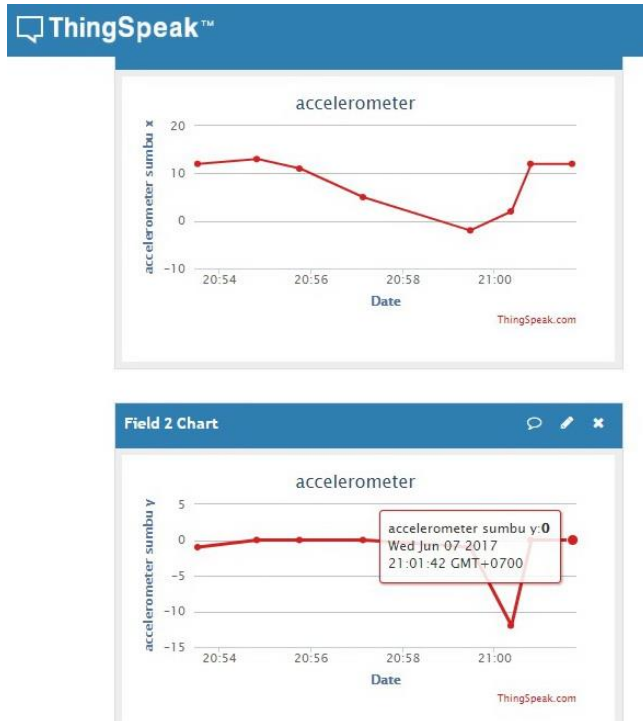
Pengiriman sensor yang terbaca pada Gambar 4.5 di *ThingSpeak* juga menunjukkan data sensor yang terbaca pada *serial monitor* dengan menunjukkan hasil data yang dibaca adalah sama, hanya saja pengiriman data yang terbaca di *serial monitor* lebih cepat terupload pada *serial monitor* daripada data sensor yang terupload di *ThingSpeak*. Dari perbandingan pernyataan diatas menunjukkan dengan *delay* 20 detik, pada pembacaan *serial monitor* dan pembacaan di *ThingSpeak* hasil data yang terbaca sesuai.

3. Dari *delay* 25 detik dengan pengambilan data sensor yang terbaca pada *serial monitor* menggunakan pewaktu *stopwatch* sebagai parameter lamanya pengiriman data yang terbaca di *serial monitor* dan dibandingkan dengan lamanya pengiriman data sensor pada *ThingSpeak*. Untuk hasil lamanya pengiriman menggunakan pewaktu *stopwatch* yang dapat dilihat pada Tabel 4.4.

Table 4.4 Pewaktu Pengiriman Data pada *Serial Monitor*

Hitungan- ke	Detik- ke	Selisih per-detik
01	00.04.69	00.04.69
02	00.35.36	00.30.67
03	01.03.23	00.27.87
04	01.29.14	00.25.91
05	01.55.14	00.26.00
06	02.26.31	00.31.17
07	02.50.96	00.24.65
08	03.19.40	00.26.44
09	03.51.91	00.27.95
10	04.20.10	00.28.19
11	04.40.08	00.19.98
12	05.04.37	00.24.29
13	05.11.98	00.07.61
14	06.04.45	00.52.47
15	06.17.91	00.13.46
16	06.34.99	00.17.08
17	06.47.48	00.12.49
18	07.15.22	00.27.74

Pembacaan data pada *ThingSpeak* dengan *delay* 25 detik menunjukkan sebagai berikut :



Gambar 4.6 Hasil Pengiriman Sensor pada *ThingSpeak* dengan *Delay* 25 detik

Pengiriman sensor yang terbaca pada Gambar 4.6 di *ThingSpeak* juga menunjukkan data sensor yang terbaca pada *serial monitor* dengan menunjukkan hasil data yang dibaca adalah sama, hanya saja pengiriman data yang terbaca di *serial monitor* lebih cepat terupload pada *serial monitor* daripada data sensor yang terupload di *ThingSpeak*. Dari perbandingan pernyataan diatas menunjukkan dengan *delay* 25 detik, pada pembacaan *serial monitor* dan pembacaan di *ThingSpeak* hasil data yang terbaca sesuai.

Perbandingan pengiriman kecepatan data sensor dari *serial monitor* dan *ThingSpeak* tentulah memiliki nilai *error*. Maka *Error* dari pembacaan sensor diatas dijelaskan pada Tabel 4.5 dan hasil *persentase* rata-rata setiap *delay* dapat dilihat pada Tabel 4.6 sebagai berikut :

Table 4.5 Data *Error* Pewaktu Pembacaan Sensor pada *Serial Monitor*

<i>Delay 15 detik</i>	<i>Delay 20 detik</i>	<i>Delay 25 detik</i>
00:40.4	00:11.6	00:04.6
01:02.1	01:00.1	01:03.2
01:24.5	01:58.2	01:55.1
03:23.5	02:54.0	02:51.0
03:49.1	03:32.1	03:51.9
04:03.3	03:50.0	04:20.1
04:24.6	05:18.2	05:04.4
05:01.7	06:30.8	06:05.5
05:46.7	07:25.2	06:35.0
06:12.9	07:54.4	07:15.2

Rumus *Error* :

Lamanya waktu pembacaan yang diperoleh - *Delay* yang ditentukan = *error*
 $\text{Error} \times 100\% / \text{delay yang ditentukan} = \text{Hasil Error (persentase)}$

Table 4.6 Hasil *Prosentase* dari *Error* Pengiriman Data

<i>Delay 15 detik</i>	<i>Delay 20 detik</i>	<i>Delay 25 detik</i>
100%	100%	100%
93,33%	95%	96%
93,33%	95%	96%
80%	90%	92%
80%	85%	88%
73,33%	85%	84%
73,33%	75%	80%
66,66%	70%	76%
66,66%	65%	76%
60%	65%	72%

Jadi hasil dari perhitungan *error* sebagai berikut :

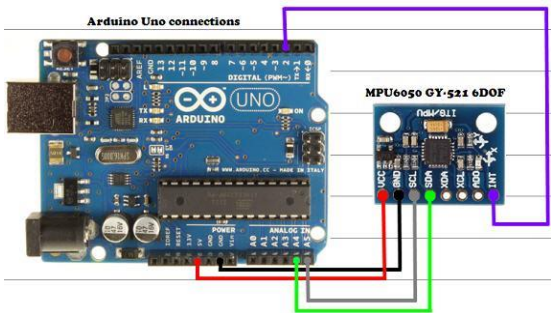
Error rata-rata dari *delay* 15 detik yaitu sebesar = 78,6 %

Error rata-rata dari *delay* 20 detik yaitu sebesar = 82,5 %

Error rata-rata dari *delay* 25 detik yaitu sebesar = 86 %

4.3 Pengujian Sensor

Pengujian sensor MPU6050 ini bertujuan untuk mengetahui apakah sensor sudah dapat bekerja dengan baik dan benar dalam mendeteksi kemiringan dalam satuan derajat. Pengujian dilakukan dengan menghubungkan sensor dan arduino lalu membandingkan dengan alat ukur busur derajat.



Gambar 4.7 Rangkaian Pengujian Sensor MPU6050

Pada Gambar 4.7 terlihat bahwa sensor dihubungkan dengan arduino melalui komunikasi *serial* I2C dimana menghubungkan pin A5 dan A4 arduino dengan pin SDA dan SCL sensor. Untuk hasil perbandingan sensor dan busur derajat ditunjukkan pada Tabel 4.7 dan Tabel 4.8.

Table 4.7 Hasil Pengujian Data Sensor Sumbu X

Sumbu X		
Data sensor (°)	Busur derajat (°)	Error Data (%)
10	10	0
20	20	0
30	30	0
41	40	2,5
57	50	14
65	60	8,3
75	70	7,1
84	80	5
86	90	4,4
Rata-rata		4,5

Table 4.8 Hasil Pengujian Data Sensor Sumbu Y

Sumbu Y		
Data sensor (°)	Besar sudut kemiringan (°)	Error Data (%)
10	10	0
20	20	0
28	30	6,6
41	40	2,5
53	50	6
68	60	13
81	70	15,7
85	80	6,2
81	90	10
Rata-rata		6,6

Berdasarkan hasil yang telah dilakukan dapat diketahui bahwa data pada sensor MPU 6050 memiliki akurasi yang tinggi pada bagian awal pembacaan derajat kemiringan. Namun untuk data selanjutnya memiliki *error*.

4.4 Pengujian Software ThingSpeak

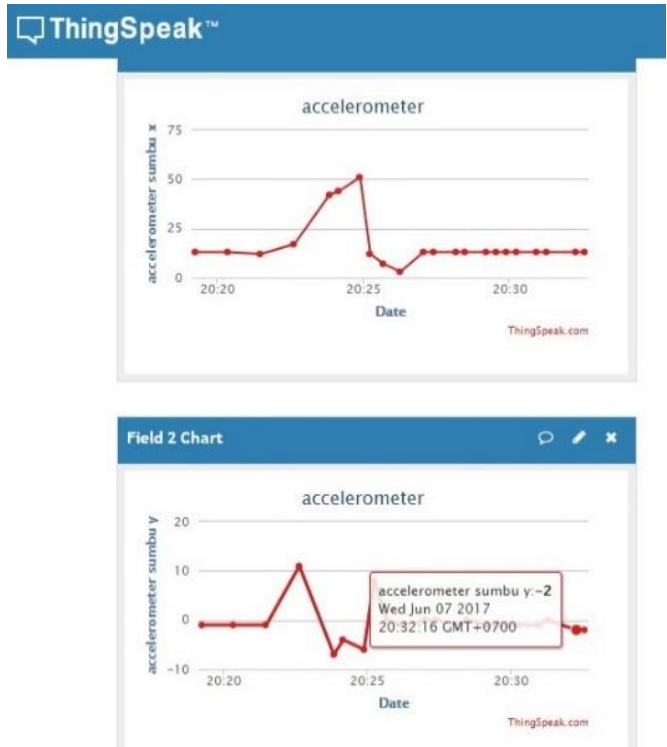
Software yang digunakan adalah menggunakan *server ThingSpeak*. Dari data sensor yang terbaca oleh *ThingSpeak* berupa tampilan grafik. Tampilan grafik ini menunjukkan kondisi saat sensor bekerja. Dari sensor *accelerometer* di gunakan parameter pengukuran bidang miring yang mempunyai nilai pada sumbu X dan sumbu Y. Sedangkan untuk sensor *proximity* digunakan parameter on-off yang terdeklarasi pada *ThingSpeak* berupa grafik angka 1 dan 0, artinya jika sensor *proximity* mengenai objek metal maka menunjukkan grafik angka 1 pada *ThingSpeak* dan jika sensor tidak mengenai objek metal maka akan menunjukkan angka 0 pada tampilan grafik di *ThingSpeak*.

a. *Delay 15 detik*

Pada sumbu X di tampilan *ThingSpeak* pada Gambar 4.8 dan sumbu Y dapat dilihat pada Gambar 4.9 menunjukkan tampilan pengiriman data sensor yang terbaca di *ThingSpeak* seperti dibawah ini :



Gambar 4.8 Hasil Pengiriman Data Sensor di Sumbu X pada *ThingSpeak*



Gambar 4.9 Hasil Pengiriman Data Sensor di Sumbu Y pada *ThingSpeak*

b. .Delay 20 detik

Pada sumbu X di tampilan *ThingSpeak* pada Gambar 4.10 dan sumbu Y dapat dilihat pada Gambar 4.11 menunjukkan tampilan pengiriman data sensor yang terbaca di *ThingSpeak* seperti dibawah ini :



Gambar 4.10 Hasil Pengiriman Data Sensor di Sumbu X pada *ThingSpeak*



Gambar 4.11 Hasil Pengiriman Data Sensor di Sumbu Y pada *ThingSpeak*

c. Delay 25 detik

Pada sumbu X di tampilan *ThingSpeak* pada Gambar 4.12 dan sumbu Y dapat dilihat pada Gambar 4.13 menunjukkan tampilan pengiriman data sensor yang terbaca di *ThingSpeak* seperti dibawah ini :



Gambar 4.12 Hasil Pengiriman Data Sensor di Sumbu X pada *ThingSpeak*



Gambar 4.13 Hasil Pengiriman Data Sensor di Sumbu X pada *ThingSpeak*

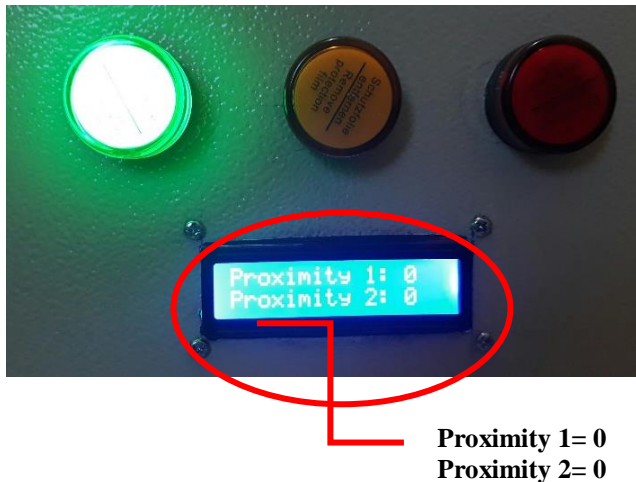
4.5 Pengujian Keseluruhan

Dari lcd yang terlihat bahwa indikator lampu hijau menyala menandakan bahwa sistem sedang bekerja yang terlihat pada Gambar 4.14, sedang untuk lcd yang terlihat bahwa indikator lampu kuning dan lampu merah menyala menandakan bahwa pengiriman data sensor ke akun *ThingSpeak* berlangsung dapat dilihat pada Gambar 4.15, Gambar 4.16 dan Gambar 4.17.

Pengujian pada *software ThingSpeak* ini dilakukan untuk melihat tampilan data sensor *proximity* sesuai dengan tampilan di layar lcd dapat dilihat pada Gambar 4.18, Gambar 4.19, Gambar 4.20 serta Gambar 4.21, hanya saja tampilan pada *ThingSpeak* memiliki *delay* dalam setiap pengiriman datanya. Hal ini dikarenakan adanya pengiriman data yang bergantung pada internet. Tetapi *ThingSpeak* mempunyai kelebihan

dapat memonitoring data sensor yang terkirim dari jarak jauh. Tampilan *ThingSpeak* berupa nilai angka 1 dan 0 yang artinya jika mengenai bimetal, sensor *proximity* akan menunjukkan nilai angka 1 dan sebaliknya. Tampilan *ThingSpeak* ini juga berupa gambar grafik yang menunjuk pada angka 1 atau 0 sesuai dengan data yang dikirim dan terlihat sewaktu hari, jam sesuai dengan data yang terkirim saat terupload pada *ThingSpeak*.

Pada Gambar 4.22 dapat dilihat bahwa dari gambar tersebut menandakan menu tampilan monitoring data sensor pada *ThingSpeak*. Dalam pelaksanaan monitoring pada *handphone android* menggunakan MIT APP *Inverter* yang berfungsi dalam melakukan monitoring sensor dari jarak jauh. Sedang untuk pengujian kecepatan Pengiriman data menggunakan *stopwatch* dengan membandingkan tampilan monitoring di layar lcd yang telah terbaca dan monitoring data sensor di *ThingSpeak* yaitu mempunyai waktu *delay* 15 detik. Hal ini dikarenakan beberapa sebab diantaranya, *jarak router wifi*, koneksi *wifi*, dan ada tidak adanya penghalang.



Gambar 4.14 Lampu Indikator Hijau Menyala saat Sistem Bekerja

Dari pengujian Gambar 4.14 merupakan pengujian pada kondisi awal dimana indikator lampu hijau menyala yang menandakan sistem aktif tetapi masih belum melakukan pengiriman data. Indikator lampu merah menandakan pengiriman data sensor *proximity* 1 dan indikator

lampu kuning menandakan untuk pengiriman data *sensor proximity* 2. Dalam tampilan lcd pada Gambar 4.14 diatas menampilkan pembacaan data sensor *proximity* 1 dan sensor *proximity* 2 belum mengenai bimetal yang artinya indikator lampu merah sensor *proximity* 1 dan indikator lampu kuning *proximity* 2 belum menyala dan menampilkan angka 0 pada tulisan *proximity* 1 dan *proximity* 2 di tampilan layar lcd.



Gambar 4.15 Lampu Indikator Merah Menyala saat Pengiriman Data Sensor

Pengujian Gambar 4.15 merupakan pengujian dimana indikator lampu hijau yang menyala menandakan sistem aktif. Indikator lampu merah menyala menandakan untuk pengiriman data sensor *proximity* 1 telah terkirim. Sedang untuk indikator lampu kuning tidak menyala karena tidak terjadi proses pengiriman data ke *ThingSpeak*. Dalam tampilan lcd pada Gambar 4.15 diatas menampilkan pembacaan data sensor *proximity* 1 setelah mengenai bimetal dengan menunjukkan angka 1 di tampilan layar lcd.



Gambar 4.16 Lampu Indikator Kuning Menyala saat Pengiriman Data Sensor

Pengujian Gambar 4.16 merupakan pengujian dimana indikator lampu hijau yang menyala menandakan sistem aktif. Indikator lampu kuning menyala menandakan untuk pengiriman data sensor *proximity* 2 telah terkirim. Sedang untuk indikator lampu merah tidak menyala karena tidak terjadi proses pengiriman data ke *ThingSpeak*. Dalam tampilan lcd pada Gambar 4.16 diatas menampilkan pembacaan data sensor *proximity* 2 setelah mengenai bimetal dengan menunjukkan angka 1 di tampilan layar lcd.



Gambar 4.17 Kedua Lampu Indikator Menyala saat Pengiriman Data Sensor

Pengujian Gambar 4.17 merupakan pengujian dimana indikator lampu hijau yang menyala menandakan sistem aktif. Indikator lampu merah dan kuning menyala menandakan untuk pengiriman data sensor *proximity* 1 dan *proximity* 2 telah terkirim. Dalam tampilan lcd pada Gambar 4.17 diatas menampilkan pembacaan data sensor *proximity* 1 dan *proximity* 2 setelah mengenai bimetal dengan menunjukkan angka 1 di kedua *proximity* tampilan layar lcd.



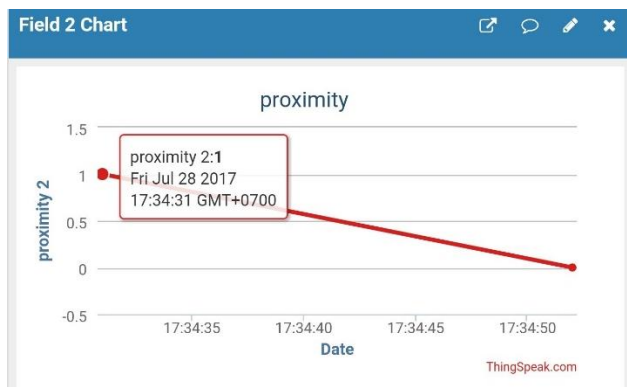
Gambar 4.18 Pembacaan Data pada ThingSpeak saat Data Sensor Terkirim

Dari Gambar 4.18 menunjukkan bahwa pembacaan data sensor *proximity* 1 tidak mengenai bimetal sehingga menampilkan gambar grafik menunjuk pada angka 0 dan data yang terbaca di *ThingSpeak* terkirim pada hari Jumat 28 Juli 2017 pukul 17.34.52.



Gambar 4.19 Pembacaan Data pada *ThingSpeak* saat Data Sensor Terkirim

Dari Gambar 4.19 menunjukkan bahwa pembacaan data sensor *proximity* 1 mengenai bimetal sehingga menampilkan gambar grafik menunjuk pada angka 1 dan data yang terbaca di *ThingSpeak* terkirim pada hari Jumat 28 Juli 2017 pukul 17.35.38.



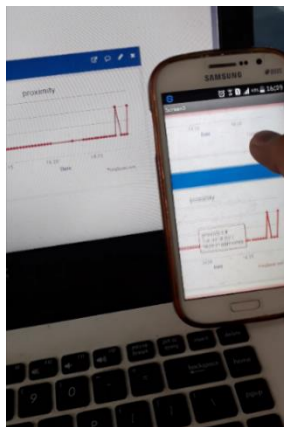
Gambar 4.20 Pembacaan Data pada *ThingSpeak* saat Data Sensor Terkirim

Dari Gambar 4.20 menunjukkan bahwa pembacaan data sensor *proximity* 2 mengenai bimetal sehingga menampilkan gambar grafik menunjuk pada angka 1 dan data yang terbaca di *ThingSpeak* terkirim pada hari Jumat 28 Juli 2017 pukul 17.34.31.



Gambar 4.21 Pembacaan Data pada *ThingSpeak* saat Data Sensor Terkirim

Dari Gambar 4.21 menunjukkan bahwa pembacaan data sensor *proximity2* tidak mengenai bimetal sehingga menampilkan gambar grafik menunjuk pada angka 0 dan data yang terbaca di *ThingSpeak* terkirim pada hari Jumat 28 Juli 2017 pukul 17.34.52.



Gambar 4.22 Tampilan Monitoring Pembacaan Data Sensor di *ThingSpeak*

Pada Gambar 4.22 menunjukkan bahwa pembacaan data sensor *proximity1* dan *proximity 2* menampilkan gambar grafik menunjuk pada angka 0 atau 1 sesuai dengan data yang terkirim. Tampilan monitoring pada *handphone* ini memiliki kelebihan yang dapat langsung mengakses data sensor sesuai dengan jenis sensornya, dan sesuai dengan pada channel di *ThingSpeak*.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil yang didapatkan selama proses pengujian dan pembuatan serta proses analisa pada Tugas Akhir ini, maka penulis mengambil beberapa kesimpulan yang berguna untuk perbaikan dan pengembangan agar nantinya dapat bermanfaat. Dari hasil pembahasan mengenai Tugas Akhir ini dapat disimpulkan bahwa:

1. Dari pengujian hasil data yang didapat menunjukkan bahwa di dalam pengiriman data sensor ke akun *ThingSpeak* dan data sensor terbaca di tampilan lcd serta menggunakan pewaktu *stopwatch* dengan mengambil sampel *delay* 15 detik, 20 detik, dan 25 detik dalam pengiriman data di *ThingSpeak* memiliki *error*. Jadi hasil dari perhitungan *error* sebagai berikut :
Error rata-rata dari *delay* 15 detik yaitu sebesar = 78,6%
Error rata-rata dari *delay* 20 detik yaitu sebesar = 82,5%
Error rata-rata dari *delay* 25 detik yaitu sebesar = 86%
2. *Error* dari pengiriman data tersebut disebabkan oleh jarak/jangkauan *wifi*, koneksi *wifi*, serta ada dan tidak adanya suatu penghalang.
3. Dari hasil uji coba sensor MPU 6050 memiliki rata-rata *error* yakni sebesar 4,5% pada sumbu X dan pada sumbu Y terdapat rata-rata *error* sebesar 6,6%.

5.2 Saran

Saran untuk penelitian selanjutnya adalah menggabungkan telemetering dan pengontrolan serta *feedback* dalam pengontrolan menggunakan aplikasi *IoT* berbasis jaringan internet *wifi*. Selain itu, juga perlu ditambahkan beberapa variasi pengujian lain pada simulasi maupun implementasi guna menguji tingkat ketelitian hasil monitoring.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Simanjuntak, Maratur G, (2013). **Perancangan Prototipe Smart Building Berbasis Arduino Uno**. Singuda ENSIKOM, Vol. 2, No. 2. Universitas Sumatera Utara, Medan.
- [2] Ahmed I. Abdul-Rahman, Corey A. Graves, Ph.D. **Internet of Things Application Using Tethered MSP430 to ThingSpeak Cloud**. IEEE Symposium on Service-Oriented System Engineering, 2016.
- [3] Virgi, A. H. (2011). **Cepat Mahir Pemograman Web Dengan Php Dan Mysql**. Jakarta: PT. Prestasi Pustakaraya.
- [4] Istiyanto,Jazi Eko. 2014. **Pengantar Elektronika dan Instrumentasi Pendekatan Project Arduino dan Android**. Yogyakarta : Penerbit Andi.
- [5] Kadir,Abdul. 2013. **Panduan Praktis Mempelajari Aplikasi Mikrokontroller dan Pemrogramannya Menggunakan Arduino**. Yogyakarta : ANDI OFFSET.
- [6] Barman,Malvino.1999. **Prinsip Elektronika Edisi ke Tiga Jilid 1**. Erlangga.Jakarta.
- [7], **MPU-6000 and MPU-6050 Product Specification Rev 3.4, Datasheet**, InvenSense, 2013.
- [8] Muhaimin. **Instalasi listrik 1**. Pusat Pengembangan Pendidikan Politeknik Bandung.
- [9] Drs.Syarifuddin Kasim, M.T., Ir.Yunus Tjandi, M.T. Program Sem Que-Batch V Teknik Elektro Relay Proteksi. **Buku Ajar Proteksi Sistem Tenaga Listrik**.
- [10],**16 x 2 Character LCD, Datasheet**,Vishay, 2002.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1 *Listing Program* pada Arduino

Program Monitoring ThingSpeak

```
#include "WiFiEsp.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
16 chars and 2 line display

/**
const int proximity_1 = 8;    // mengubah 'buttonSaya' menjadi variabel
pin 2
const int proximity_2 = 9;    // mengubah 'buttonSaya' menjadi variabel
pin 2
const int led_hijau = 4;      // mengubah 'ledSaya' menjadi variabel pin
3
const int led_kuning = 3;     // mengubah 'ledSaya' menjadi variabel
pin 3
const int led_merah = 2;     // mengubah 'ledSaya' menjadi variabel pin
3
int kondisi_1 = 0;           // mengubah 'buttonStatusSaya' menjadi variable
0
int kondisi_2 = 0;
int y,z;
//
// Emulate Serial1 on pins 6/7 if not present
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7); // RX, TX
#endif

char ssid[] = "AUDY";        // your network SSID (name)
char pass[] = "wonosari1204"; // your network password
int status = WL_IDLE_STATUS; // the Wifi radio's status

String apiKey = "WGWA64E0E9UXYP73";
char server[] = "api.thingspeak.com";
```

```

// Initialize the Ethernet client object
WiFiEspClient client;
//*****

//*****
void setup()
{
  lcd.init();           // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
//*****
  pinMode(led_hijau, OUTPUT); // mengubah 'ledSaya' menjadi pin
OUTPUT
  pinMode(led_kuning, OUTPUT); // mengubah 'ledSaya' menjadi pin
OUTPUT
  pinMode(led_merah, OUTPUT); // mengubah 'ledSaya' menjadi pin
OUTPUT
  pinMode(proximity_1, INPUT); //mengubah 'buttonSaya' menjadi pin
INPUT
  pinMode(proximity_2, INPUT); //mengubah 'buttonSaya' menjadi pin
INPUT

  // initialize serial for debugging
  Serial.begin(115200);
  // initialize serial for ESP module
  Serial1.begin(9600);
  // initialize ESP module
  WiFi.init(&Serial1);

  // check for the presence of the shield
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue
    while (true);
  }

  // attempt to connect to WiFi network
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");

```

```

Serial.println(ssid);
// Connect to WPA/WPA2 network
status = WiFi.begin(ssid, pass);
}

// you're connected now, so print out the data
Serial.println("You're connected to the network");

printWifiStatus();

Serial.println();
Serial.println("Starting connection to server...");
// if you get a connection, report back via serial

}

void loop()
{

    kondisi_1 = digitalRead(proximity_1);
    kondisi_2 = digitalRead(proximity_2);
    z = 0;
    if (kondisi_2 == HIGH)
    {
        digitalWrite(led_kuning, HIGH); //maka 'ledSaya' menyala
        y=0;
    }
    else
    {
        digitalWrite(led_kuning, LOW); //apabila tidak ada, maka 'ledSaya'
        mati
        y=1;
    }

    if (client.connect(server,80)) { // "184.106.153.149" or
    api.thingspeak.com
    if (kondisi_1 == HIGH){
        digitalWrite(led_merah, HIGH); //maka 'ledSaya' menyala
        String postStr = apiKey;

```

```

postStr += "&field1=";
postStr += String(z);
postStr += "&field2=";
postStr += String(y);

```

```

client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);

```

```

lcd.setCursor(0,0);
lcd.print("Proximity 1:");
lcd.setCursor(13,0);
lcd.print(z);

```

```

lcd.setCursor(0,1);
lcd.print("Proximity 2:");
lcd.setCursor(13,1);
lcd.print(y);

```

```

}
else
{

```

```

    digitalWrite(led_merah, LOW); //apabila tidak ada, maka 'ledSaya'

```

mati

```

    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(z+1);
    postStr += "&field2=";
    postStr += String(y);

```

```

client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");

```

```

    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

    lcd.setCursor(0,0);
    lcd.print("Proximity 1:");
    lcd.setCursor(13,0);
    lcd.print(z+1);

    lcd.setCursor(0,1);
    lcd.print("Proximity 2:");
    lcd.setCursor(13,1);
    lcd.print(y);
  }
}
client.stop();
Serial.println("Waiting...");
// thingspeak needs minimum 15 sec delay between updates
delay(15000);

}

void printWifiStatus()
{
  // print the SSID of the network you're attached to
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your WiFi shield's IP address
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

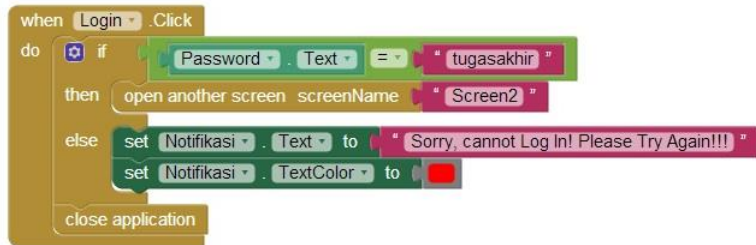
  // print the received signal strength
  long rssi = WiFi.RSSI();
  Serial.print("Signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}

```

-----Halaman ini sengaja dikosongkan-----

A.2 Listing Diagram Blok MIT APP Inventor

Blok Tampilan Login



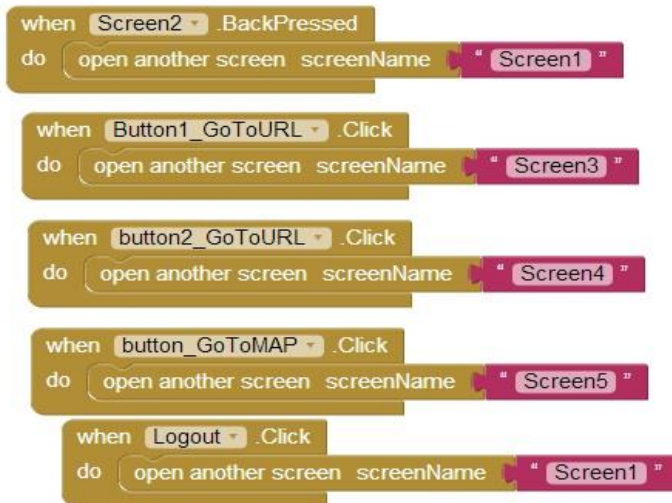
Blok Screen beberapa pilihan, diantaranya:

Go To URL Proximity

Go To URL Accelerometer

Go To MAP

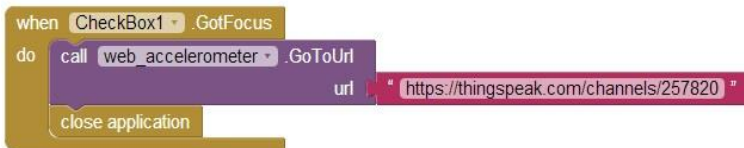
Logout



Tampilan untuk menampilkan web monitoring sensor Proximity



Tampilan untuk menampilkan web monitoring sensor Accelerometer



Tampilan untuk menampilkan halaman Web pada akses MAPS PT PAL Surabaya



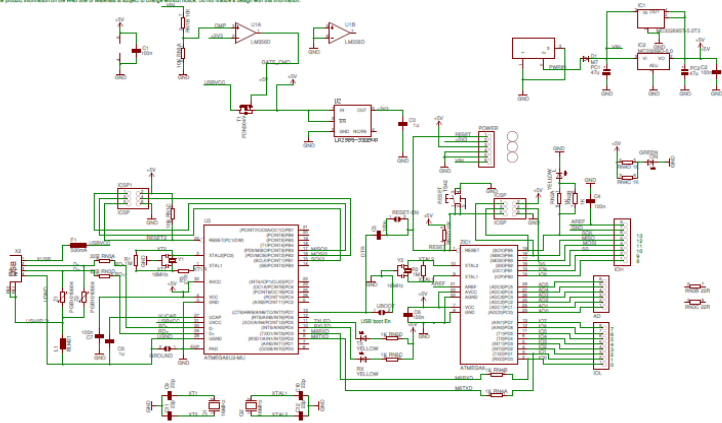
LAMPIRAN B

B.1 *Datasheet* Arduino Uno

Arduino™ UNO Reference Design

Warranty Descriptions are PROVIDED "AS IS" AND, WITH ALL FAULTS, ANY IMPLIED WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Autodesk may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Autodesk reserves the right to alter specifications at any time. Autodesk does not warrant, either expressly or implied, that the product information on this Web Site or Materials is accurate, complete or free of errors. Customer should not rely on the information. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with the information.



Technical Specification

EAGLE files: [aridino-damilarova-unc-dsdsz.zip](#) Schematic: [aridino-unc-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The Input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C:** 4 (SDA) and 5 (SCL). Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#).

B.2 Datasheet ESP8266 wifi shield

1. Preambles

ESP-12E WiFi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna.

The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller.

ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

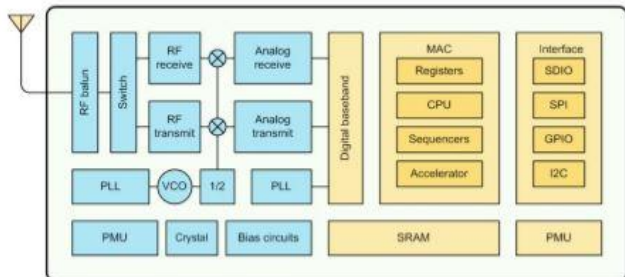


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.1. Features

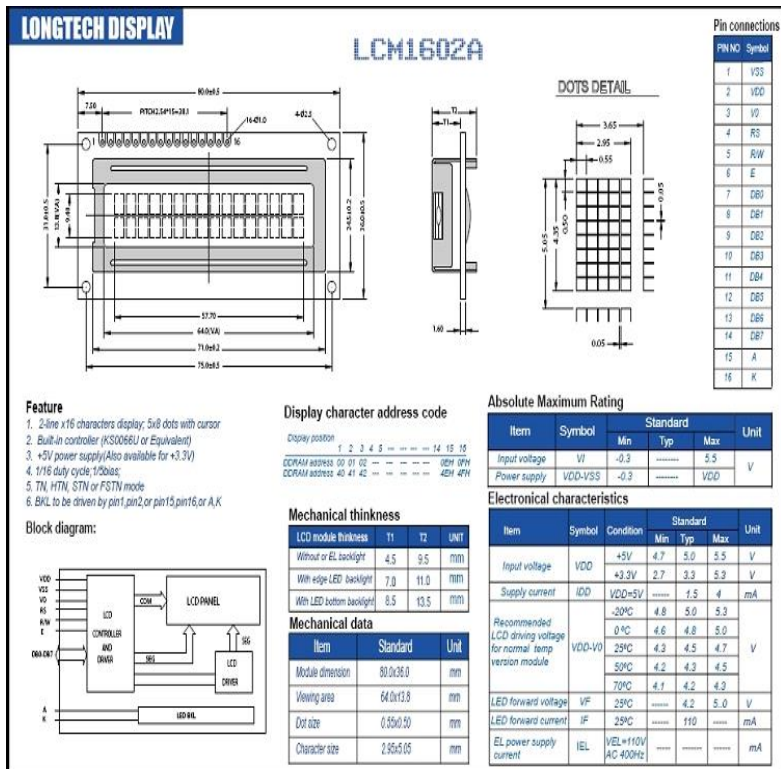
- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation and 0.4s guard interval



- Deep sleep power $< 10\mu\text{A}$, Power down leakage current $< 5\mu\text{A}$
- Wake up and transmit packets in $< 2\text{ms}$
- Standby power consumption of $< 1.0\text{mW}$ (DTIM3)
- $+20\text{dBm}$ output power in 802.11b mode
- Operating temperature range $-40^{\circ}\text{C} \sim 125^{\circ}\text{C}$

-----Halaman ini sengaja dikosongkan-----

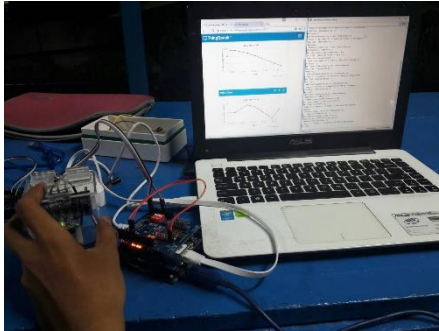
B.3 Datasheet LCD 16x2



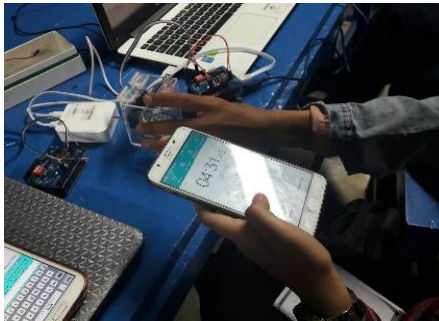
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C PENGUJIAN ALAT

C.1 Pengujian Kecepatan Pengiriman Data



Gambar C.1 Pembacaan Sensor di ThingSpeak



Gambar C.2 Penghitungan Lamanya Waktu Pembacaan Data antara *Thingspeak* dan *Serial Monitor*

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Audy Zahra Firdausy
Jenis Kelamin : Perempuan
Agama : Islam
TTL : Surabaya, 19 April 1995
Alamat Rumah : Jl. Wonosari Wetan
Baru XII-A /4 Surabaya
Telp/HP : 085748925642
E-mail : audyzahra95@gmail.com
Hobi : Berenang

RIWAYAT PENDIDIKAN

- 2001 – 2007 : SDN WONOKUSUMO VII/46 Surabaya
- 2007 – 2010 : SMPN 5 Surabaya
- 2009 – 2013 : SMAN 8 Surabaya
- 2014 – sekarang : Bidang Studi Elektro Industri, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PT DELTA JAYA MAS , Gresik (Juni-Juli 2015)
- Kerja Praktek di PLN Area Pamekasan, Pamekasan (Juli-Agustus 2016)

PENGALAMAN ORGANISASI

- Staff Departemen Dalam Negeri BEM FTI ITS (2015-2016)
- Staff Departemen Kesejahteraan Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro ITS (2015-2016)